ERDC/CHL SR-09-6

Coastal and Hydraulics Laboratory

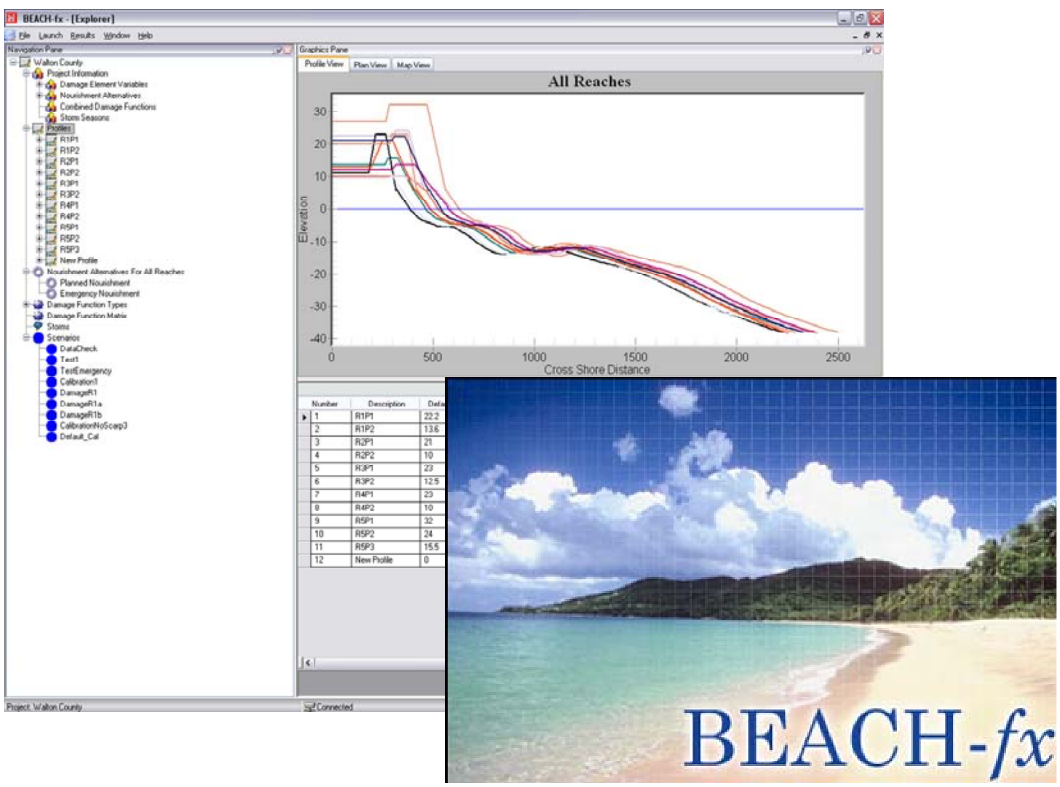
US Army Corps of Engineers®
Engineer Research and Development Center

*Flood and Coastal Storm Damage Reduction Program*

# Beach-*fx* User's Manual: Version 1.0


Cory M. Rogers, Kristine A. Jacobson, and Mark Gravens


August 2009



*Flood and Coastal Storm Damage*
*Reduction Program*

ERDC/CHL SR-09-1
August 2009

# Beach-*fx* User's Manual: Version 1.0

Cory M. Rogers and Kristine A. Jacobson

*CDM*
*1050 N. Reed Station Road*
*Carbondale, IL 62902-7377*

Mark B. Gravens

*U.S. Army Engineer Research and Development Center*
*Coastal and Hydraulics Laboratory*
*3909 Halls Ferry Road*
*Vicksburg, MS 39180-6199*

Final Report



Prepared for U.S. Army Corps of Engineers
Washington, DC 20314-1000

**Abstract:** The need to strengthen the linkages between engineering analyses (project performance and evolution) and planning functions (alternative analysis and economic justification) with respect to coastal storm damage reduction projects within the Corps has led to the development of the life-cycle simulation model Beach-*fx*. Beach *fx* provides a comprehensive analytical framework for evaluating the physical performance and economic benefits and costs of shore protection projects, particularly beach nourishment along sandy shores. The model has been implemented as an event-based Monte Carlo life-cycle simulation tool that is run on desktop computers. This report describes the components, purpose, and operational function of the Beach-*fx* graphical user interface, including navigation within the interface and the organization and specification of all model input and output data.

# Contents

# Figures and Tables

## Figures

## Tables

# Preface

The work described herein was authorized by Headquarters, U.S. Army Corps of Engineers (HQUSACE). Funding was provided by the Flood and Coastal Storm Damage Reduction Research Program (FCSDR) and the Shore Protection Assessment Research Program (SPA), administered by the U.S. Army Engineer Research and Development Center (ERDC), Coastal and Hydraulics Laboratory (CHL), Vicksburg, MS. Technical Program Director for the FCSDR was Dr. Jack E. Davis, ERDC-CHL. William R. Curtis, ERDC-CHL, was Program Manager for FCSPR and SPA. The research that led to the development of Beach-*fx* was conducted by ERDC-CHL, and the U.S. Army Engineer Institute for Water Resources (IWR), and contractors at CDM under contract number W912HQ-04-0007. Dr. David A. Moser, IWR, provided the original vision for the development of Beach-*fx*.

The research conducted by CDM Federal Programs, Inc. was performed under the general supervision of Cory M. Rogers. The research conducted at CHL was performed under the general supervision of Ty B. Wamsley, Chief, Coastal Processes Branch, Bruce A. Ebersole Division Chief, Coastal Flood and Storm Protection, Thomas W. Richardson, former Director CHL, and Dr. William D. Martin, Director, CHL.

COL Gary E. Johnson was Commander and Executive Director of ERDC. Dr. James R. Houston was Director.

# 1 Introduction

Beach-*fx* is designed to assist users in evaluating and analyzing the benefits and costs of hurricane protection and storm damage reduction projects. The following provides a basic User's Manual for the Beach-*fx* engineering-economic planning tool. Beach-*fx* was developed through a collaborative effort between the Institute of Water Resources (IWR) and the U. S. Army Engineer Research and Development Center (ERDC). The goal of this guide is to provide a basic set of instructions on the common model functions, data entry and editing steps necessary to utilize Beach-*fx* in a shore protection study.

An overview of the background, nature, functions, and application of Beach-*fx* is provided in supplemental overview material (Gravens et al. 2007; Moser et al. 2007; Males et al. 2007).

It is recommended that the user review this material if not already familiar with Beach-*fx*. Review of the appropriate National Economic Development (NED) manuals is further recommended for an understanding of the planning processes involved.

## Installation

The Beach-*fx* software is distributed via Internet download. The installation files can be obtained from the Beach-*fx* Web site at http://hera.pmcl.com/beachfx. Upon request to ERDC, the application can be obtained on CD-ROM as well.

## System requirements

Before installing Beach-*fx*, ensure that the target computer system meets the minimum required hardware and software requirements:

| | |
|---|---|
| Processor | Pentium 4 or AMD Athlon XP processor |
| Operating System | Windows XP |
| Memory | 512 MB RAM minimum |

| | |
|---|---|
| Hard Disk | 1 GB hard disk space |
| Peripherals | Internet connectivity or CD-ROM drive |
| System Conditions | Administrator privileges on the target computer (if you are utilizing Microsoft Windows NT, Windows 2000, or Windows XP) |

# 2 Installation Instructions

## Internet download installation

1. Download the installation file to a temporary directory on the target computer (example: C:\Temp).
2. Click the **Start** button on the Taskbar.
3. Click **Run**.
4. Type the complete path to the downloaded file (example: C:\Temp\Beachfx_Install.exe).
5. Click the **OK** button.
6. Wait for the setup program to start.
7. Follow on-screen instruction to complete the setup.

## CD-ROM installation

1. Insert the Beach-*fx* CD-ROM into your CD-ROM drive.
2. A Web page outlining the contents of the CD should launch automatically.
3. Click the link titled **Beach-*fx* Application Version 1.0**.
4. Wait for the setup program to start.
5. Follow on-screen instruction to complete the setup.

## Uninstall procedures

1. Click the **Start** button on the Taskbar.
2. Click **Settings**.
3. Click **Control Pane**l.
4. Double-click **Add/Remove Programs**.
5. Select **Beach-*fx***.
6. Click the **Add/Remove button**.

# 3 Interface Basics

The Beach-*fx* application is a Windows-based, menu-driven Multiple Document Interface (MDI) application. An MDI application allows for multiple windows to be opened within the main application window. This enables Beach-*fx* to display alternate views of information simultaneously on the screen (i.e., multiple graphs and/or reports open at the same time for easy comparison of results). The basic user view is a combination of a menu system and a three-pane view, consisting of a standard tree-based logical hierarchy of study information in one pane, a data grid reflecting information about the selected item in the hierarchy in a second pane, and a third pane representing, where appropriate, a graphic display of the information selected.

The system as a whole consists of the user interface, the associated databases and input/output files and a computational Kernel or simulation engine, a separate program that is launched from the user interface to perform the Monte Carlo simulation. Each such run of the Monte Carlo simulation is a named Scenario, typically a set of run parameters specifying simulation duration, number of iterations and associated conditions, and specifications for the run, such as particular parameters for Planned Nourishment and Emergency Nourishment. The user interface allows for inspection of the output associated with a given Scenario.

## Menu system

The menu system used in Beach-*fx* is modeled closely after other familiar Windows® applications. Figure 1 displays the complete menu system found in the top portion of the screen when Beach-*fx* is running. Figure 2 displays the import menu and Figure 3 displays the export menu.

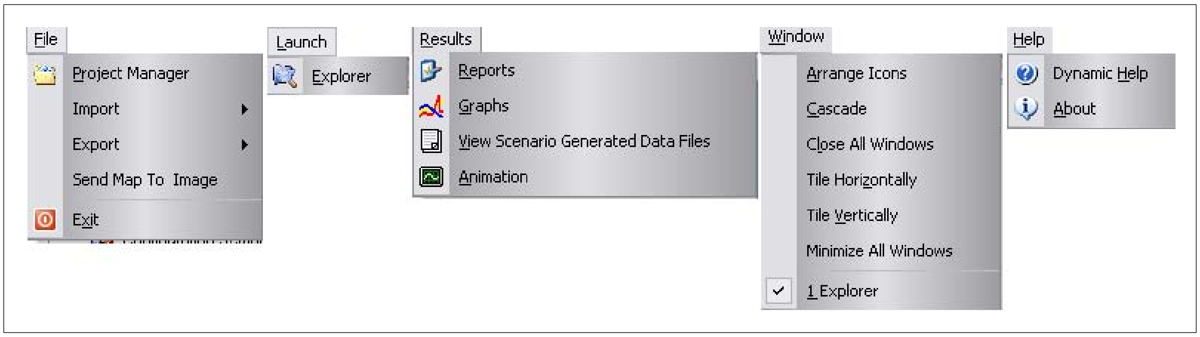



Figure 1. Beach-*fx* main menu system.

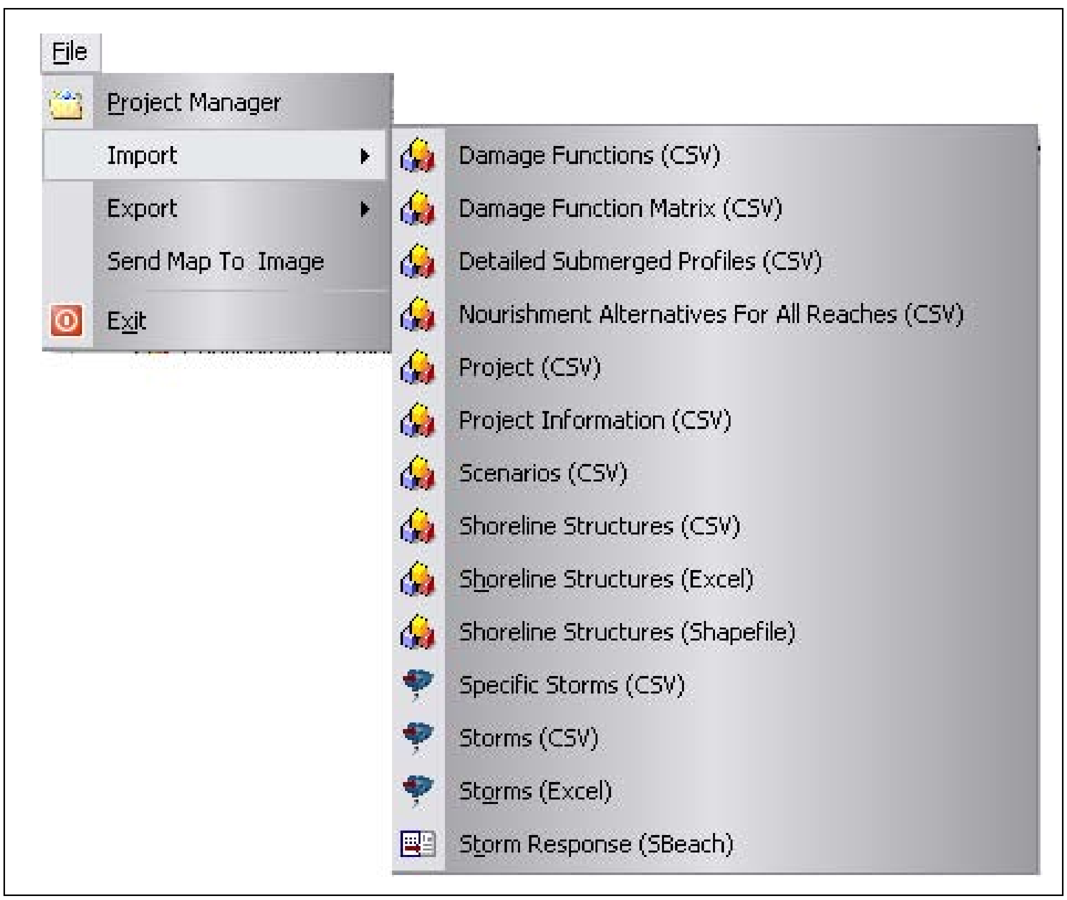



Figure 2. Beach-*fx* import menu.

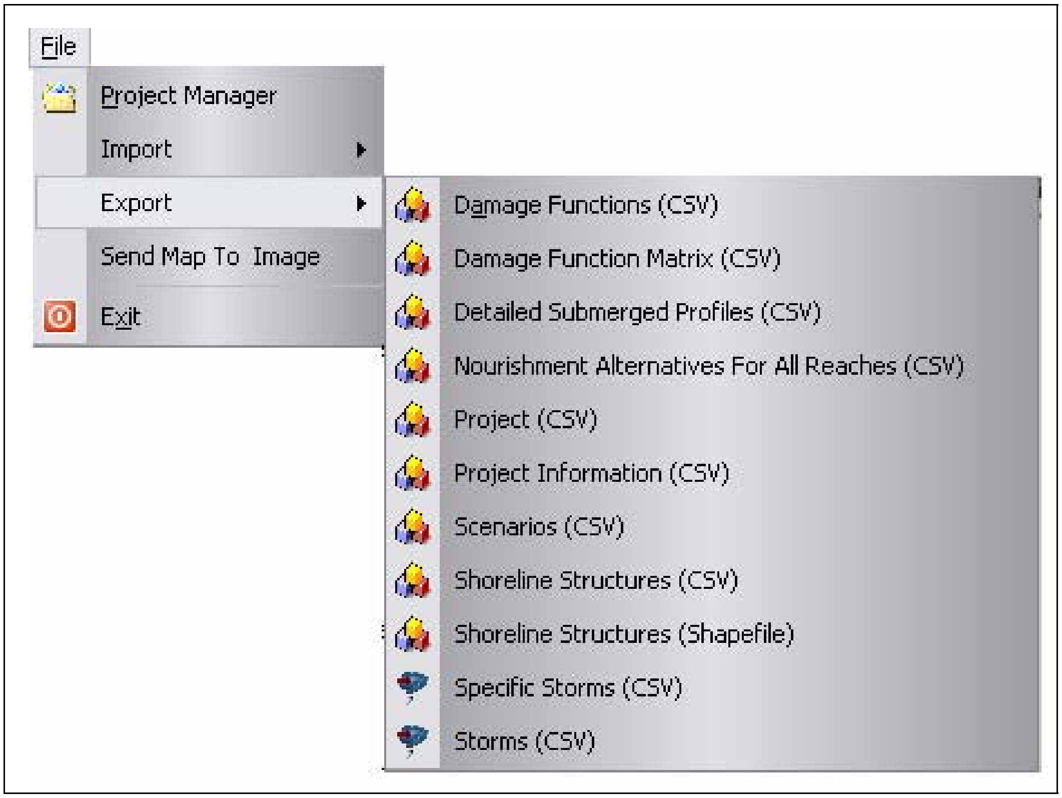



Figure 3. Beach-*fx* export menu.

## Data organization

Beach-*fx* uses data from a number of sources. The user interface is the control center for organizing, viewing, and editing this information. The basic organizing concept is that of the project, a combination of Microsoft® Access databases that represent the study area being examined.

The basic files that are managed through the user interface are:

1. The master Microsoft® Access database, named Governor.mdb, used to store general information on the projects that are being examined.

2. The individual study Microsoft® Access databases containing inputs and outputs. There are three such databases that must be associated with each project, identified by filename suffixes:

*a*. Input database (IDB extension). Containing the primary information about the study area (Profiles, Reaches, Damage Functions, Damage Elements, Lots, Nourishment Alternatives, and Storms).

*b*. Shore response database (SDB extension). Containing the results of process model runs (e.g., SBEACH) defining the profile response to storms contained in the IDB, including morphology change and wave, water elevation, and erosion profiles.

*c*. Output database (ODB extension). Used to store the outputs from the simulation.

It is possible to combine individual instances of these three databases in different projects, i.e., the same ODB can be shared by multiple projects, if desired, but, in general, there will be a unique set of three databases (IDB, ODB, and SDB) for a given project.

3. Data files generated by SBEACH simulations, used to create the SDB. These files are external from the interface/project files and are required only to build the SRD. Once the SDB is populated they are no longer needed by Beach-*fx*.

4. Import data files, in comma-separated value (CSV) format, used to populate the IDB database.

5. Import data files, in Microsoft Excel format, used to populate Storm, Profile, Reach, Lot, and Damage Element information in the IDB database.

6. Import data files, in shapefile format, used to populate Reach, Lot, and Damage Element information in the IDB database.

7. Output data files containing detailed results associated with a simulation run, in CSV and plain (ASCII) text format.

The user interface provides the ability to organize these files to work on a particular project and examine the results.

If Beach-*fx* is launched but is unable to locate or connect to a study database (Governor.mdb), all menu section headers and corresponding sub-menu items will be disabled. If a successful connection is established but no current project exists, the File -> Project Manager menu option will be left enabled to allow users to create a project from either existing data (IDB, ODB, and SDB) or from scratch using the provided database templates.

### File menu

**Project manager**. Opens the Project Manager form. Used for the Creation, Deletion, or Modification of the database files associated with a shore protection study.

**Import**. Menu sub-section header (see Chapter 9 for more details).

- **Damage functions (CSV)**. Imports Damage Functions from a CSV template file.
- **Damage function matrix (CSV).** Imports the Damage Function Matrix into Beach-*fx* from a CSV template file. NOTE: Data will only be imported for the Matrix in overwrite mode.
- **Detailed Submerged Profiles (CSV).** Imports X, Y submerged profile data points from a CSV template for any/all Profiles.
- **Nourishment Alternatives for all Reaches (CSV).** Imports defined Nourishment Alternatives (Planned, Emergency, and Planform Rates) for any/all Reaches.
- **Project (CSV).** Imports data for an entire project from CSV templates. The method reads each template file name and location from a generated INI file.
- **Project Information (CSV).** Allows the user to import Beach-*fx* Project Information data. These data appear under the Project Information header node in the Explorer's *Navigation Tree*. It includes Damage Element Variables (Armor, Construction, Damage Element, and Foundation Types), Nourishment Alternatives, Storm Seasons, Output Levels, and Configuration Settings.

- **Scenarios (CSV).** Imports Scenario data from a CSV import template file.

- **Shoreline structures (CSV).** Imports shoreline structure data (Profiles, Reaches, Lots, and Damage Elements) from CSV template files.

- **Shoreline structures (Excel).** Imports pre-existing shoreline structure information from Microsoft® Excel spreadsheet templates, including Profiles, Reaches, Lots, and Damage Elements.

- **Shoreline structures (shapefile).** Imports shoreline structure data (Reaches, Lots, and Damage Elements) from Geographical Information System (GIS) Shapefile templates.

- **Specific storms (CSV).** Imports Specific Storm data from a CSV import template file.

- **Storms (CSV).** Imports historical storms from a CSV import template file.

- **Storms (Excel).** Imports historical storms from a Microsoft® Excel spreadsheet template file.

- **Storm response (SBEACH).** Used for importing storm response data from SBEACH output files.

**Export.** Menu sub-section header (see Chapter 9 for more details).

- **Damage functions (CSV).** Exports the current project's Damage Function data to a generated CSV file.

- **Damage function matrix (CSV).** Exports the current project's Damage Function Matrix to a generated CSV file.

- **Detailed Submerged Profiles (CSV).** Exports the current project's Detailed Submerged Profile data to a generated CSV file.

- **Nourishment Alternatives for all Reaches (CSV).** Exports selected Nourishment Alternative data for the current project's Reaches to generated CSV files. Options include Planned Nourishment

Alternatives, Emergency Nourishment Alternatives, and Reach Planform Rates.

- **Project (CSV).** Exports all data for the current project to generated CSV files. This will also generate an INI file that can be read by the Beach-*fx* project import method, which specifies the name, location, and type of contained data for each of the exported files.

- **Project Information (CSV).** Exports selected Project Information data for the current project. These data appear under the Project Information header node in the Explorer's *Navigation Tree* and include Damage Element Variables (Armor, Construction, Damage Element, and Foundation Types), Nourishment Alternatives, Storm Seasons, Output Levels, and Configuration Settings.

- **Scenarios (CSV).** Exports Scenario data for the current project to a generated CSV file.

- **Shoreline structures (CSV).** Exports data for selected shoreline structures (Profiles, Reaches, Lots, and Damage Elements) to generated CSV files.

- **Shoreline structures (Shapefile).** Exports data for selected shoreline structures (Reaches, Lots, and Damage Elements) to generated GIS Shapefiles.

- **Specific storms (CSV).** Exports the current project's Specific Storm data to a generated CSV file.

- **Storms (CSV).** Exports the current project's Storm data to a generated CSV file.

**Send map to image.** Allows the user to save the Beach-*fx* Map Pane's current display to an image file (Bitmap, GIF, or JPEG). This menu option is disabled if the Map Pane is not the selected pane (i.e., not being viewed) or if the map does not contain any visible layers.

### Launch menu

**Explorer.** Used to open the main application window for displaying/editing a shoreline protection study. There are two instances where the user may

need to invoke this menu option: (1) if the Governor.mdb being used by the Beach-*fx* executable does not contain a defined current project on startup (this may happen the first time the application is run) or (2) if the user accidentally closes the *Study Explorer* window within the Beach-*fx* GUI.

### Results menu

**Reports.** Used for selection and generation of output reports.

**Graphs.** Used for selection and generation of graphical outputs.

**View scenario generated data files.** Allows users to select and display any files that were generated during the execution of a defined Scenario. Users may also choose to remove all outputs for a specific Scenario, which removes all related records from the ODB as well as deletes all related data files.

**Animation.** Allows users to run an animation depicting changes in coastal morphology for a specific Scenario. Only Scenarios that have been run with the *MorphologyTimeLine* Output Option checked can be animated.

### Window menu

**Arrange icons.** Used to arrange all minimized windows within the application.

**Cascade.** Used to resize all open windows to equal size and place them over the top of one another, offsetting each window enough so that each window's caption is visible. Clicking on one of the captions brings that window to the front of the screen.

**Close all windows.** Used to close all of the system's open windows.

**Tile horizontal.** Used to resize all open windows so that each window has the same width as the application and then line up each window onto one another.

**Tile vertical.** Used to resize all open windows so that each window has the same height as the application, then place each window side-by-side.

**Minimize all windows.** Used to minimize all of the system's open windows and arrange them along the bottom of the application window.

### Help menu

**Dynamic help.** Opens an electronic copy of the *Beach-fx User's Manual*.


Figure 4. Beach-*fx* information screen.

**About.** Displays information about Beach-*fx* such as current versions for the application, databases, and simulation Kernel, that can be used to identify information needed when application support is required (Figure 4).

## Status bar

Figure 5 shows an example of the Beach-*fx* status bar located at the bottom of the application window. Its primary purpose is to supply visual queues to the user pertaining to the current project, data connectivity, and versioning.


Figure 5. Beach-*fx* status bar.

The following is a listing and explanation for each panel found in the status bar.

**Current project.** Displays the name of the project that is currently open.

**Connection.** Displays the *Connected* icon when a valid database connection is present and the *Disconnected* icon when no database is currently selected.

**Application version.** Displays the current version of the Beach-*fx* application being used.

**Time.** Displays the current system time.

**Date.** Displays the current system date.

## Study Explorer

Figure 6 shows an example of the *Study Explorer* which is the main screen for all data entry and data manipulation in the system. The *Study Explorer* is divided into three individual windows or panes: *Navigation Pane*, *Graphics Pane*, and *Data Entry Pane*. On the left edge of the screen is the *Navigation Pane*. The *Navigation Pane* categorizes and organizes all of the study data elements into a hierarchical view of information. The pane located in the top right section of the screen is the *Graphics Pane*. The *Graphics Pane* will display graphical representations of study data, such as Profiles, Reaches, Lots, and Damage Elements. In the lower right section of the display is the *Data Entry Pane*. The *Data Entry Pane* allows for the input or editing of all study information through a spreadsheet style grid. This three-pane approach provides access to all necessary data through a single portal. To further facilitate and simplify data entry, all three panes are linked together and will focus on a single data element at any given point in time. For example, clicking on a particular data item (such as a Lot) in the *Navigation Pane* will highlight the specific Lot in the *Graphics Pane* and display the associated Lot information in the *Data Entry Pane*. All three panes can be resized using the standard Windows sizing controls located in the upper right corner of each pane for a customized look and feel suited to each particular user.

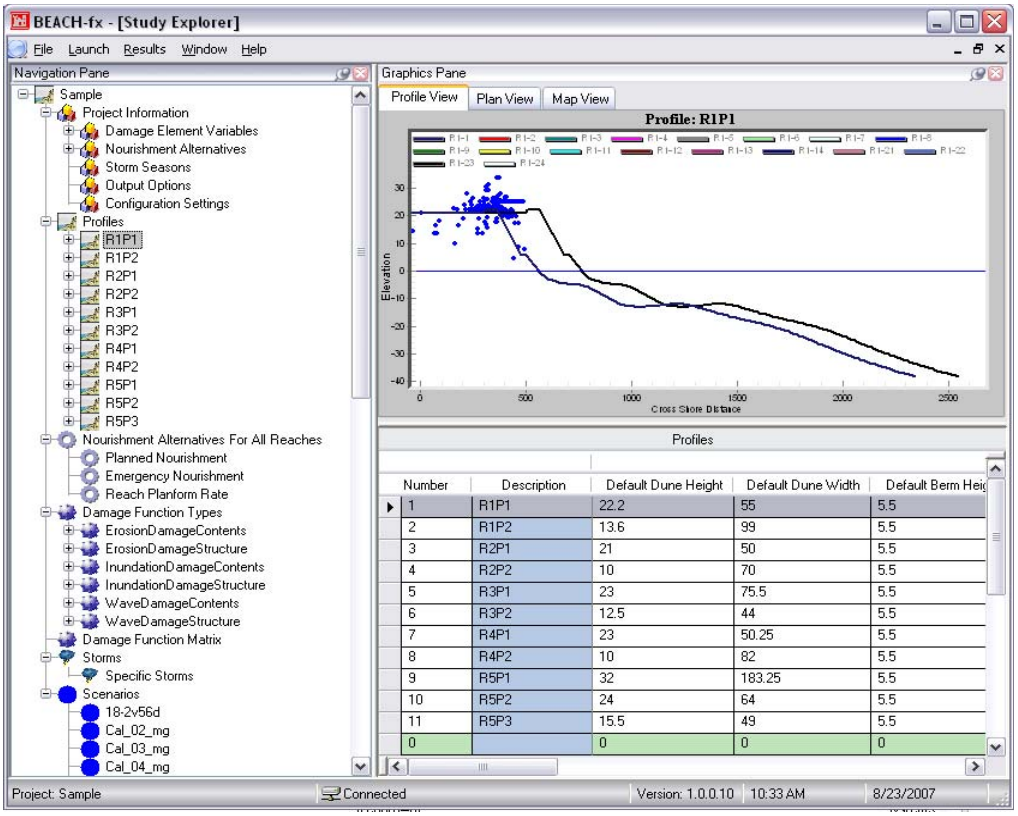



Figure 6. Beach-*fx* study explorer.

## Navigation Pane

The *Navigation Pane* uses a hierarchical view type display to organize study information. This allows the data to be placed in an organized fashion for easy identification and retrieval of information. For example, with a quick glance at the *Navigation Pane* you can easily decipher which Damage Elements belong to a particular Lot or which Reaches are associated with a particular Profile.

An example of the *Navigation Pane* is shown in Figure 7. Each type of item has its own unique icon, which helps to further distinguish the different data elements. Within Figure 7, some items, like the Reach, Reach 1, contain child items under them. This allows for access to the associated information for that Reach (in this example, the Planned Nourishment Alternatives or Lots). Simple nodes, such as Damage Elements, do not have subordinate or child nodes as no other data items are grouped under a particular Damage Element.

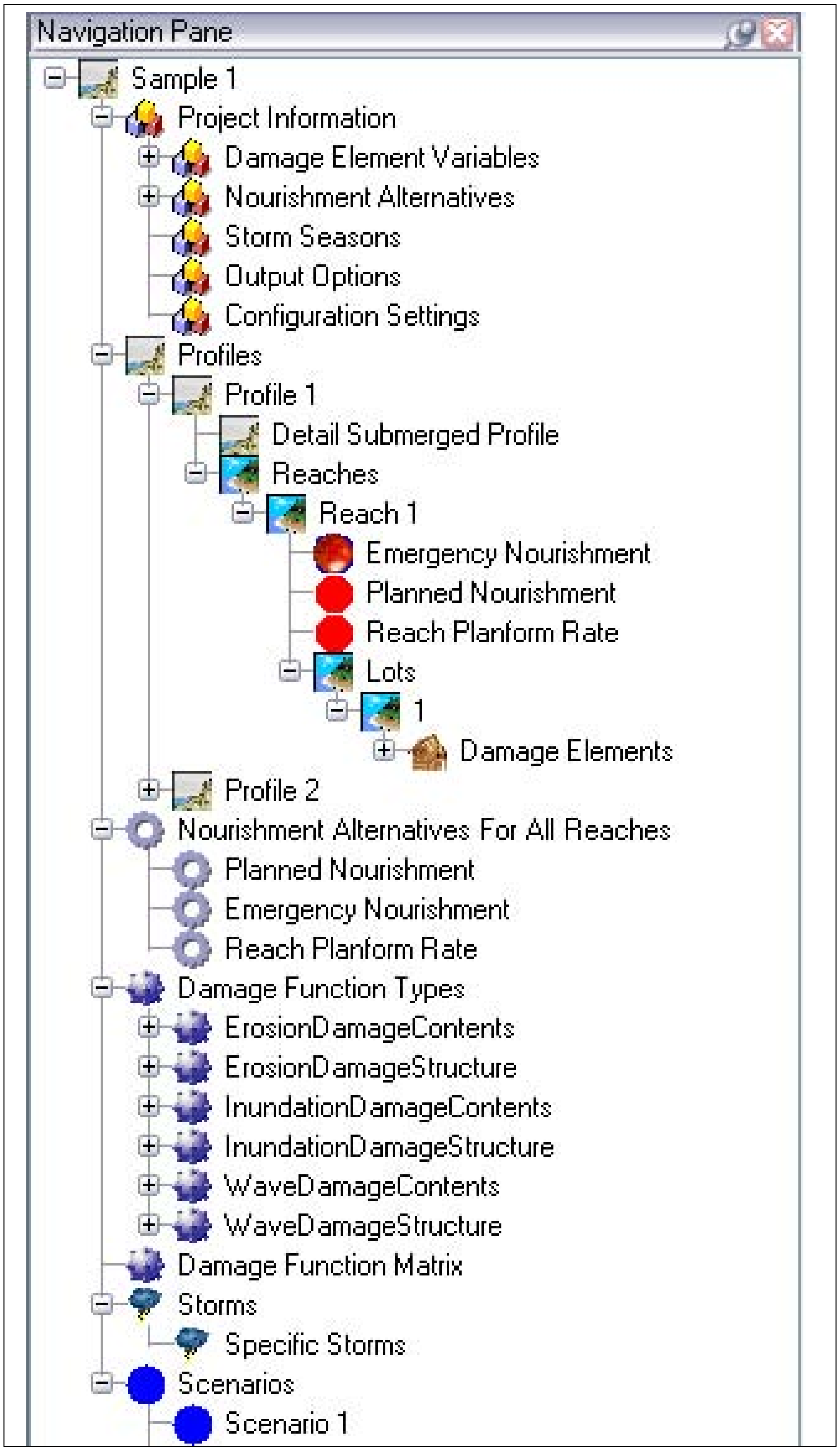



Figure 7. Navigation Pane.

The *Navigation Pane* can be used as an alternative to the *Data Entry Pane* for adding and deleting some information. There are also a number of options for importing/exporting data into and out of a Beach-*fx* project

(see Chapter 9). Some of these options are context specific and are therefore not available under the application's file menu. This additional functionality is provided through the use of a context menu, which is a menu that appears when you right mouse click on an item in the *Navigation Pane*. An example of this menu can be seen in Figure 8. Context menus are available for all nodes in the *Navigation Tree*. However, the menu options contained within each menu depends on the type of object node clicked.

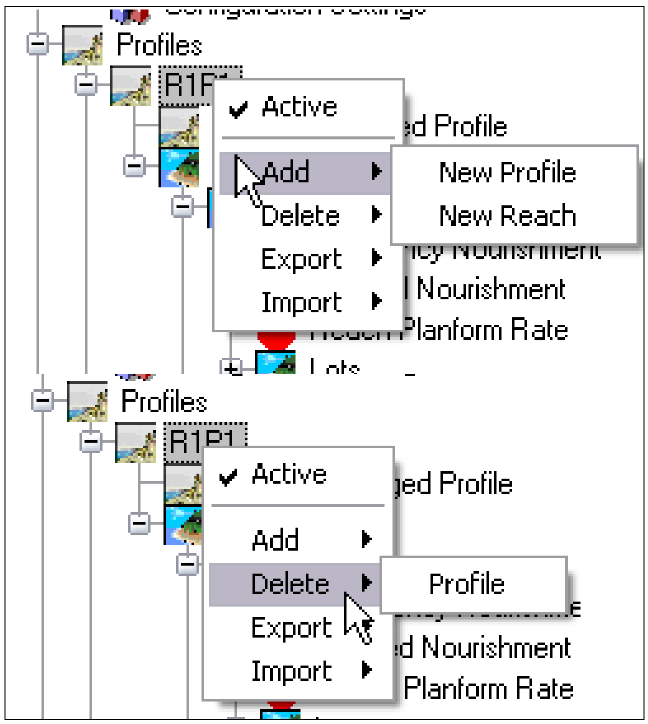



Figure 8. Context sensitive menus.

## Data Entry Pane

The *Data Entry Pane* shown in Figure 9 is used for all data entry necessary for successful population of the Beach-*fx* IDB database. It utilizes a spreadsheet/grid to display and edit data. The grid also allows for many customizations to ease the process of data entry such as sorting, rearranging, and filtering of columns.

When an item is selected in the *Navigation Pane*, the *Data Entry Pane* displays all related items of the associated type in its grid, with the corresponding item selected and/or colored to distinguish it from the other data. For example, selecting the Reaches node below a specific

Profile in the *Navigation Pane* will display all Reaches belonging to that Profile in the data grid. In this manner, users can view data by object type and parent relationship (where it exists).

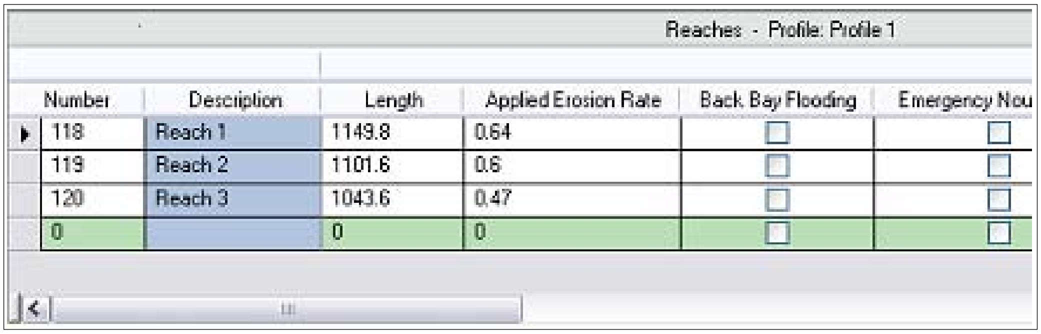



Figure 9. Data entry pane.

## Adding data

Required study information can be entered through the *Data Entry Pane*. A new item can be added by entering information into the blank row at the bottom of the grid, which is colored green to distinguish it from records that have already been saved to the database. Entering data into the blank row will cause a new object record to be inserted into the appropriate table in the database and a new object node will be added to the *Navigation Pane*, if one is required for that particular item type. Some object grids contain required fields (highlighted in blue). It is important to note that, no matter how much information is entered into the blank row, the new record will not be saved until data have been entered for those fields. For instance, in the preceding example (Figure 9), the Description Column is a required field. This happens to be the only required column in the grid, so once a description as been entered into the empty row of the grid, a new Reach will be created and a corresponding node will be created in the *Navigation Tree*.

There are some grids for which new records cannot be added. For instance, neither the Output Options nor Configuration Settings tables contain an empty row. The entries in these two tables are dictated by the Kernel and cannot be added or deleted by the user. In addition, most of the fields are highlighted gray, indicating they are read-only and the contents cannot be edited. The Damage Function Matrix is another example (see Chapter 5, “Damage Functions” for more details). In these types of grids, only non-grayed fields can be modified.

Many of the object grids contain a Number or External ID field. These fields are used to uniquely identify object records and maintain any established hierarchical relationships when data are outside of the database (i.e., when data have been exported). In an effort to help the user, the application is designed to calculate and insert the next available number into this field when a new record is created if the user has not manually entered a value. Of course, users are free to enter their own values into these fields. However, if a value is entered that already exists for another record, a message will display a warning that the value is non-unique and therefore not valid. In Figure 10, the user has attempted to assign the same Number value for Reach 4 as Reach 3. The warning is displayed and once the user has acknowledged it by pressing the **OK** button, the field's value will revert back to the last valid entry.

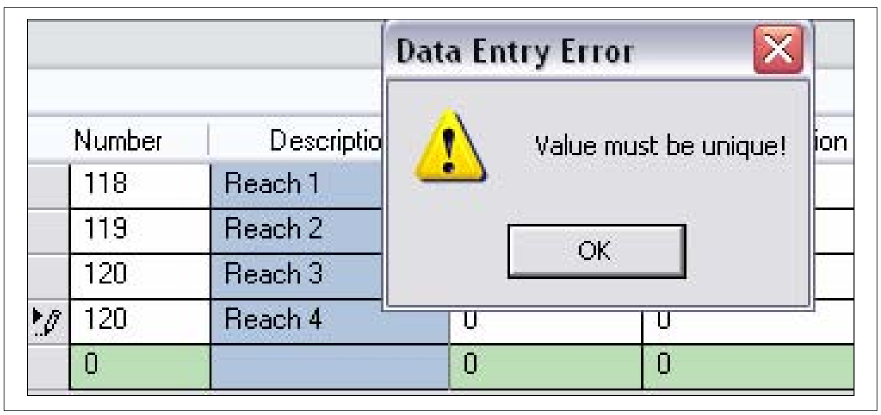



Figure 10. Non-unique data entry error.

## Deleting data

To delete an item through the *Data Entry Pane*, highlight the particular item record(s) in the data grid to be deleted and press the **Delete** key on the keyboard. This will bring up a delete confirmation dialog, as shown in Figure 11, to allow for confirmation or cancellation of the item deletion. Once the deletion of the item has been confirmed, the item will be removed from the system and the database. Deleted records cannot be restored and must be reentered if the deletion was inadvertent or premature.

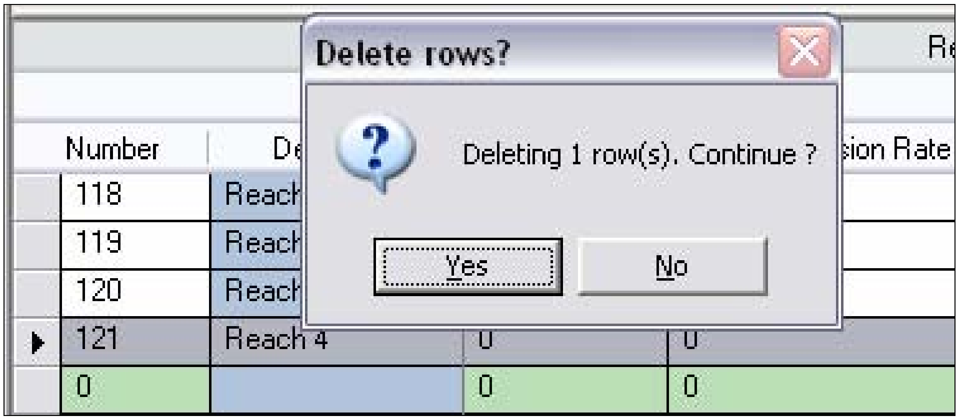



Figure 11. Delete confirmation.

## Graphics Pane

The *Graphics Pane* is used to view graphical representations of various data elements (i.e., Reaches, Lots, and Damage Elements). The pane contains three different tab options: *Profile*, *Plan*, and *Map View*. The display for each tab option is dictated by what object node is currently selected in the *Navigation Pane*.

### Profile View

Profiles are plotted in cross-section, Reach-by-Reach in this view. An example of the *Graphics Pane* displaying all study Reaches in graphical form is shown in Figure 12.

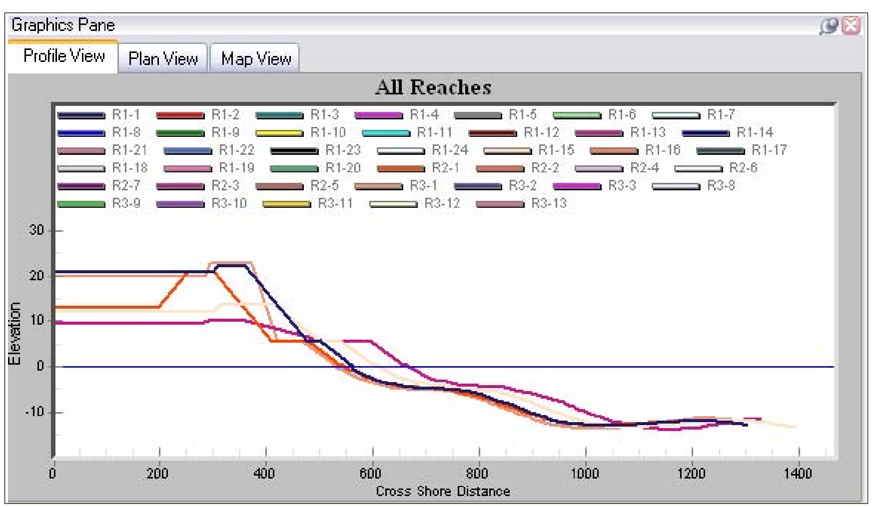



Figure 12. *Profile View* for all Reaches.

Figure 12 is an example of the display that is created when the Profiles header node is selected in the *Navigation Tree*. The horizontal line marking elevation zero is the user-defined datum. The portion of the graph above the datum is drawn based on defined Profile data input such as dune height, dune width, berm height, berm width, dune slope, etc., in conjunction with the upland width defined for each child Reach. The portion of the graph below the datum or the submerged profile is plotted in one of two different ways depending on whether a Profile is flagged as having a Detailed Submerged Profile. If this flag is set for a Profile, the Profile will have a Detailed Submerged Profile child node in the *Navigation Tree* and it is expected that the user has provided the set of X, Z data points to be used to define the submerged portion of the profile. If the flag is not set, an equilibrium profile is created based on the user specified sediment scale parameter.

If a specific Profile node is selected in the *Navigation Tree*, only those Reaches belonging to that Profile will be displayed (Figure 13). Moreover, any child Damage Elements will be plotted. If a specific Reach is selected, only that Reach will be drawn, and only the Damage Elements belonging to that Reach's child Lots will be plotted. In this way, the display appropriately reflects the relational hierarchy of the shoreline objects. If the user drills down to the Damage Element level and selects a specific Damage Element node in the *Navigation Tree*, the selected Damage Element will be highlighted red to differentiate it from any others that might be present in the same Lot (Figure 14).

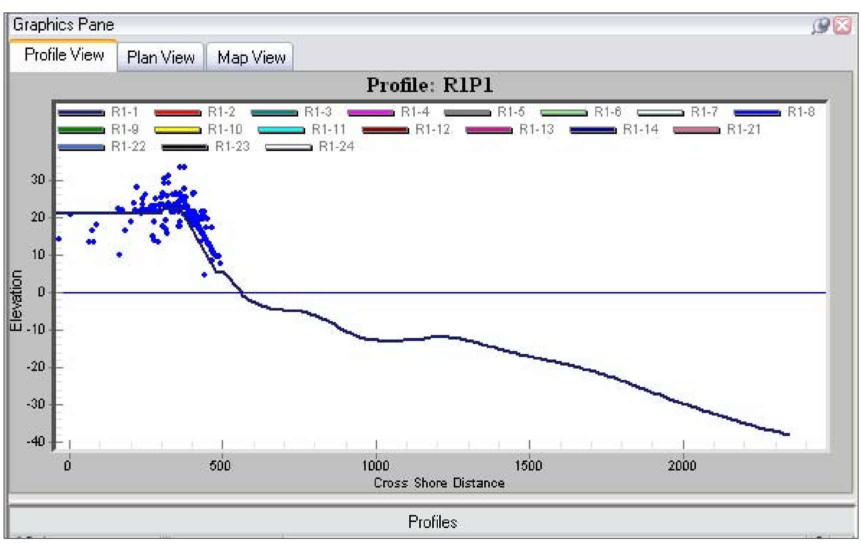



Figure 13. *Profile View* for specific Profile.

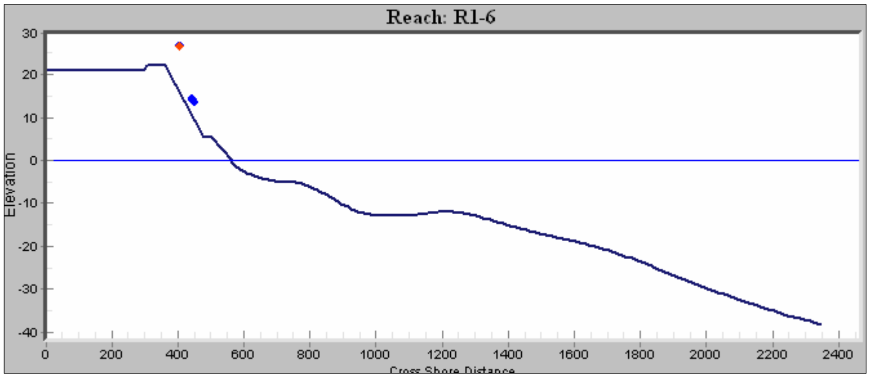



Figure 14. *Profile View* for selected Damage Element.

## Plan View

Users can view a graphical representation of shoreline objects (Reaches, Lots, and Damage Elements) from an aerial perspective using the *Graphics Pane's Plan View*. As in the *Profile View*, selecting the Profiles header node in the *Explorer's Navigation Tree* will draw all Reaches in the project (Figure 15).

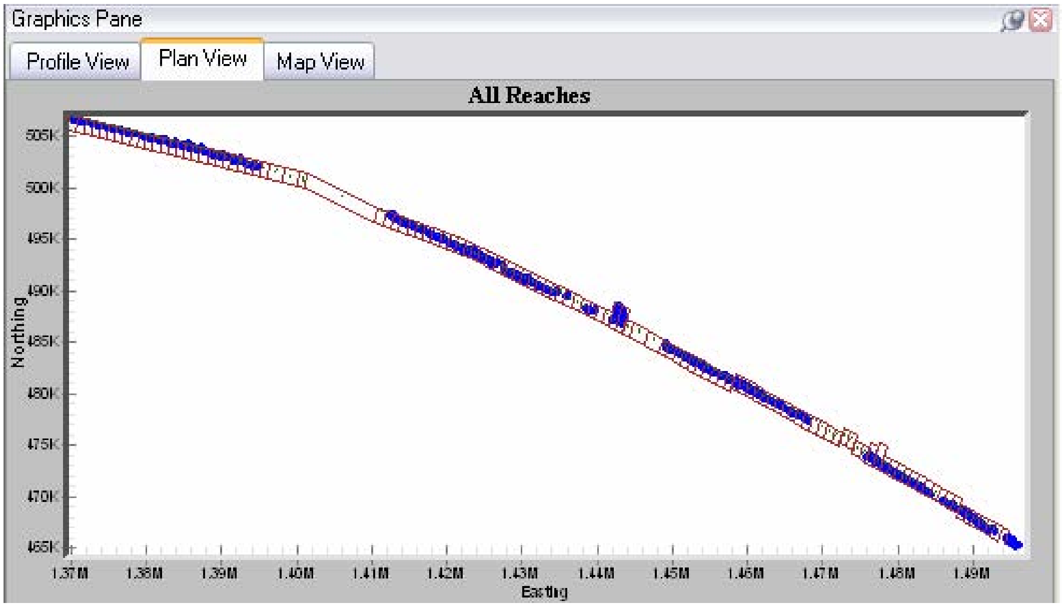



Figure 15. Plan View for all Reaches.

This display shows all Reaches in the current project, which results in a complete representation of the project layout with the bottom left corner being the seaward direction and the top right corner being the landward

direction. Finer detail is achieved by drilling down into the hierarchical model. A specific Reach is displayed by selecting its node in the *Navigation Tree* and a Lot by selecting its node and so on. Likewise, selecting a particular Profile will display all Reaches for that Profile.

Figure 16 shows the Reach R1-1, drawn in red, with each of the Reach's six Lots outlined in black. The blue dots within the Lots are the Damage Elements reference. Again, the bottom edge is the seaward direction and the top edge is landward. The SBEACH line is drawn in green. Drilling into one of the Lots and clicking on a Damage Element node causes the display to change as illustrated in Figure 17.

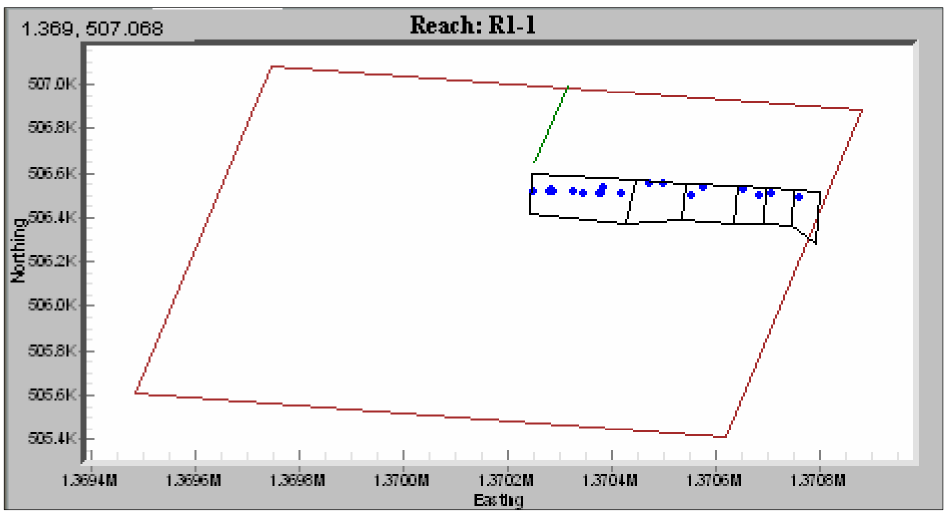



Figure 16. Plan View for all Lots for Reach R1-1.

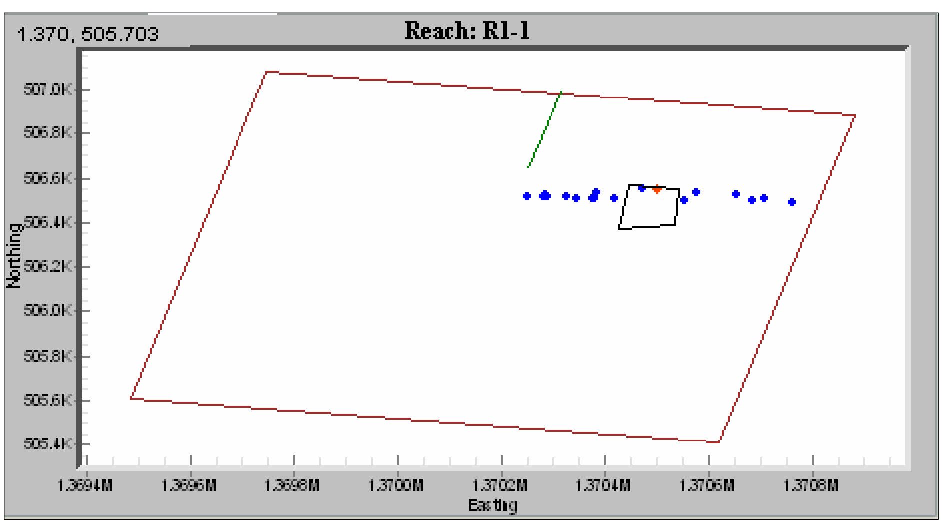



Figure 17. Plan View for selected Damage Element.

### Map View

The *Map View* tab on the *Graphics Pane* is used to display GIS files associated with a particular project. This is done through the Beach-*fx* *Project Manager* by specifying the location of a GIS directory containing only shape and image files (MrSID) related to an individual project. These files are then loaded into the map display. An example of the *Map View* is displayed in Figure 18.

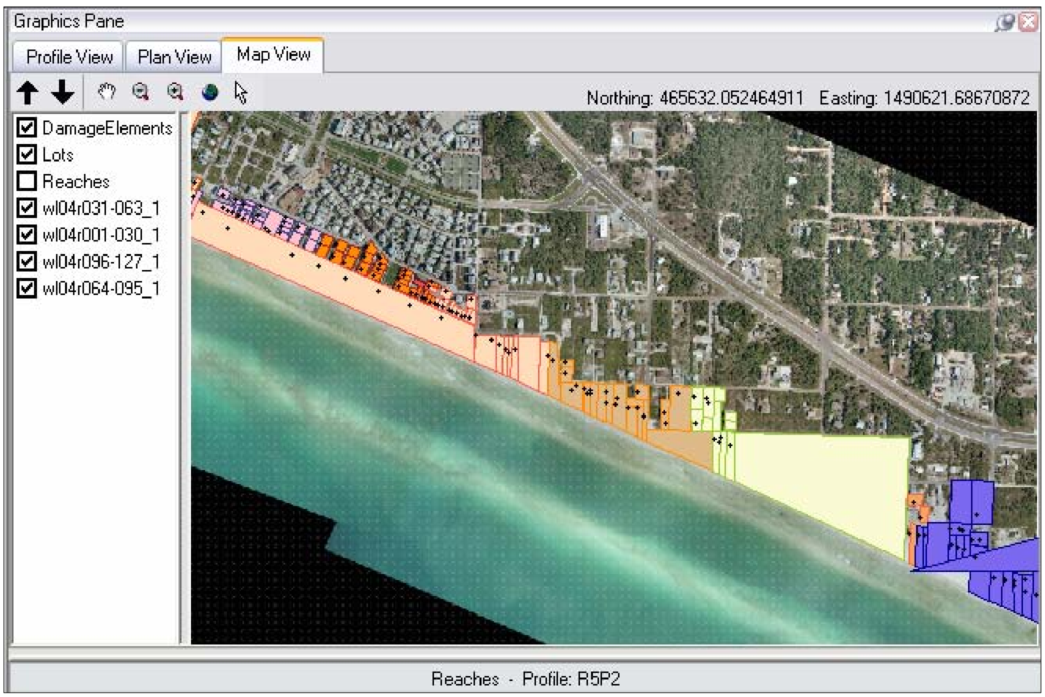



Figure 18. Beach-*fx* map pane.

Like other applications that display GIS data, such as ESRIs ArcGIS, each file included in the Beach-*fx* *Map View* is rendered as a separate layer. Each layer is included in the list on the left side of the map image and can be made visible or invisible by checking/un-checking the corresponding checkbox. In Figure 19, the Reaches layer has been turned off. Map layers are drawn in the order of bottom to top, meaning that the first layer added to the map will be the bottom layer and display underneath any other layers. By design, Beach-*fx* attempts to determine the most appropriate default layer ordering. The application looks at the files present in the specified GIS directory and adds any image files to the map first so that they will comprise the bottom layers. Shapefiles are added next with files containing polygons being added before files containing points. Only valid

ESRI shapefiles can be read in by the Beach-*fx* application. The assumption is that Damage Elements should be drawn on top of Lots, which should in turn be drawn on top of Reaches. For this reason, the shapefile names (case insensitive) are taken into account. Files with names containing reach or reaches are assumed to be Reach shapefiles, files with names containing lot or lots are assumed to contain Lot data and files with names containing damage or de are assumed to contain Damage Element data. If the default ordering is not correct, users can reorder the map layers by selecting a layer entry in the list and clicking the ↑ or ↓ buttons in the map's toolbar (Figure 19).

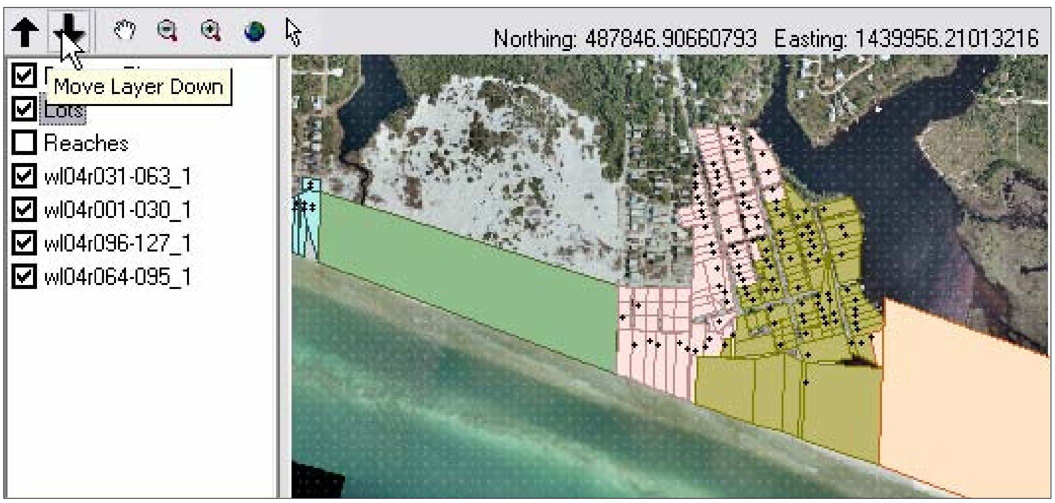



Figure 19. Reordering map players.

Clicking the **Globe** button will set the map extent to the combined extents of all visible layers. Users can also pan the map image, zoom in or zoom out by selecting the appropriate tool from the toolbar. It is possible to save the current map display to an image file (Bitmap, JPEG, GIF) by selecting **File -> Send Map To Image** from the application's main menu. This same feature is available by right-clicking on the map to bring up the map's context menu, shown in Figure 20. This right-click feature is only available if no other map tool is selected, meaning the mouse cursor is the standard arrow cursor, rather than the panning hand for example. Users can switch from a chosen map tool back to the selection cursor by clicking the button in the map toolbar containing the arrow cursor icon.

If the shapefiles used to create the display are representative of the shoreline objects (Reaches, Lots, Damage Elements) defined in the database and if the shapefiles contain attribute tables with the third field

storing the external IDs for those objects (i.e., Reach Number, Lot, External ID, Damage Element External ID), the map will reset its extent automatically to focus on a shoreline object selected by the user in the *Explorer's Navigation Tree*. For instance, in Figure 20, the user has clicked on the Lots node beneath a specific Reach. In the map image, the Lots for this Reach are displayed in yellow. The map's extent is calculated internally to be that of the parent Reach's, automatically focusing the user on the correct location. Note also that all child Lots of this Reach are rendered with a heavier outline, distinguishing them from Lots belonging to other Reaches present in the display.

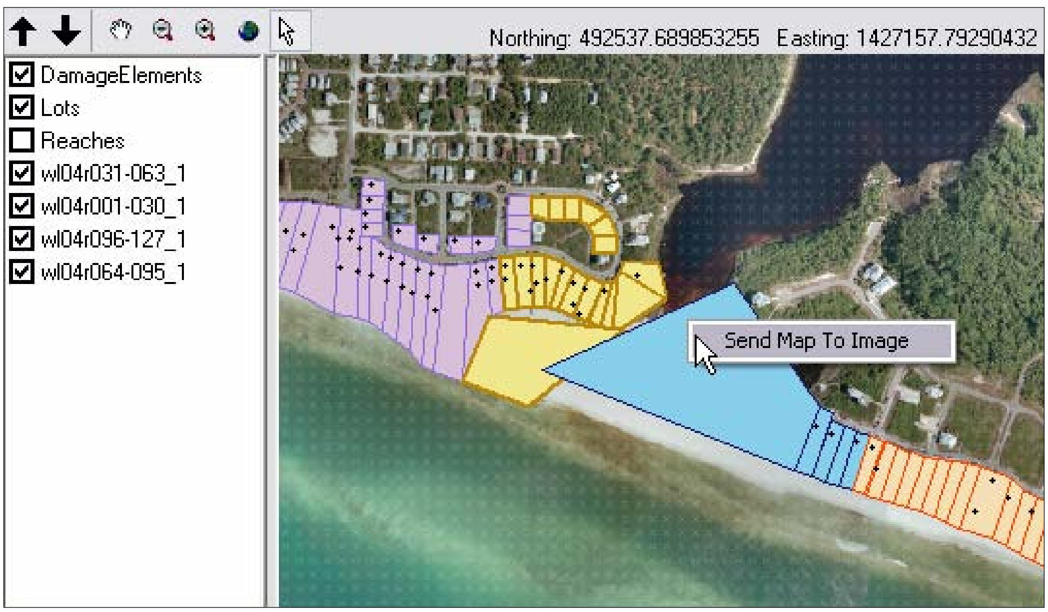



Figure 20. Map display context menu.

## Project Manager

The *Project Manager* is the central location used for managing project files and is shown in Figure 21. It is accessible through the application's main menu by selecting **File -> Project Manager** or if a project is open, by right-clicking on the project header node in the *Navigation Tree* and selecting the option from the corresponding context menu. Operations performed by the *Project Manager* utility are broken into three categories: **Open**, **Create**, and **Manage**. The specific functions performed by each utility are as follows:

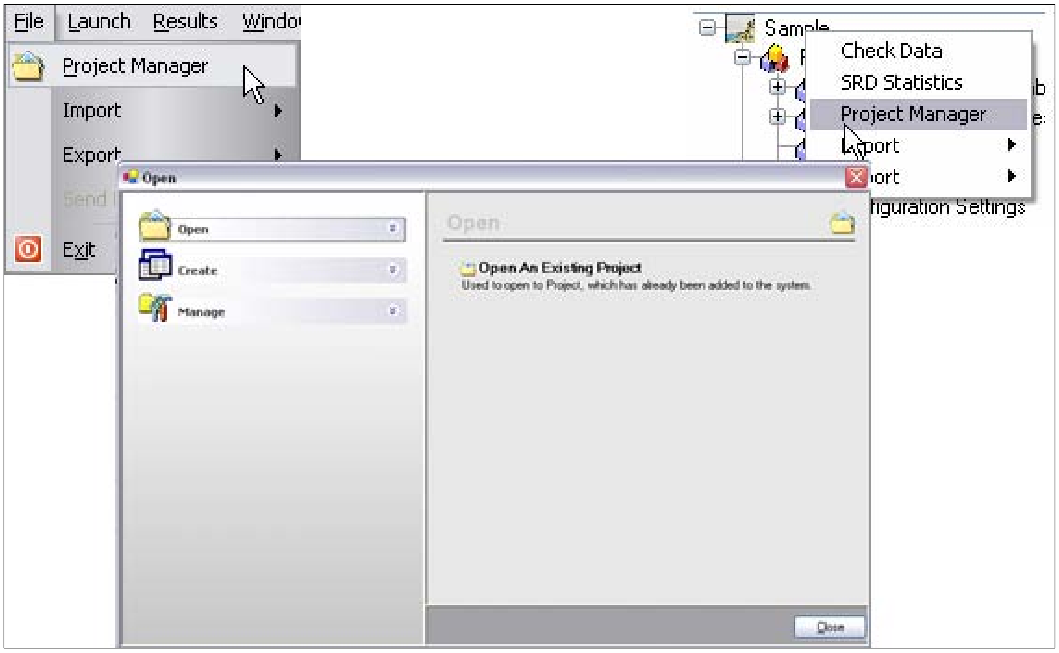
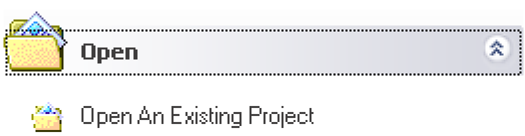


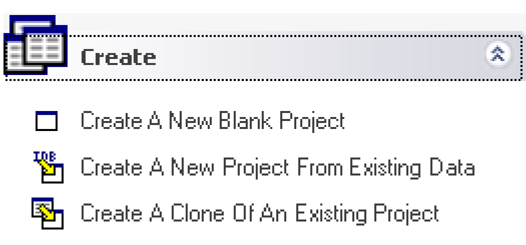

Figure 21. Beach-*fx* Project Manager.

### Open




**Open an existing project.** Used to locate and open an existing shore protection project file. When the *Project Manager* is first opened, this is the default screen displayed. It contains a drop-down list of all existing, user-defined projects. Users can open a different project by selecting the name of the desired project from the list and clicking the **Open** button.

### Create




**Create a new blank project.** Used to add a new project to the system. The user is required to enter a name for the new project and provide a

location on the system hard drive where the project files will be created (Figure 22).

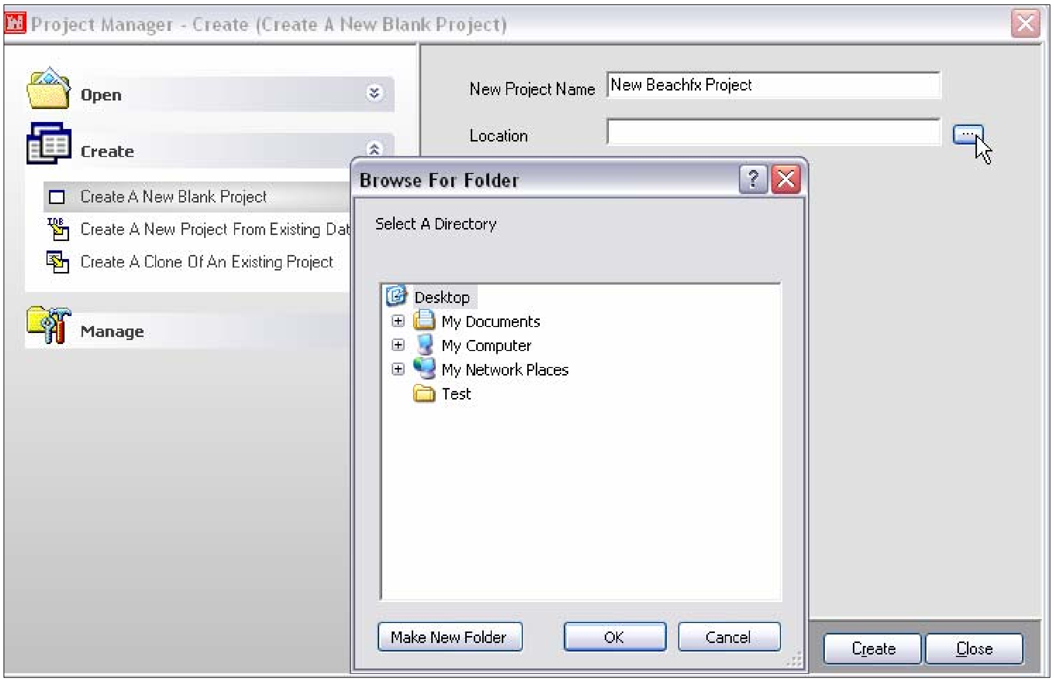



Figure 22. Project Manager to create new project.

**Create new project from existing data.** Used to add an existing project to the system, for which the required databases have been previously generated. This capability is useful for sharing project files among the team members. The project's Input (IDB), Output (ODB), and SRD (SDB) databases are required to complete this step (as shown on the form in Figure 23). Users have the option to specify a GIS directory where project-related shape and image files are located. However, specification of a GIS directory is not necessary for the project to be created. The entry can be left blank and as long as the database locations are specified, the project will be created with an empty map display.

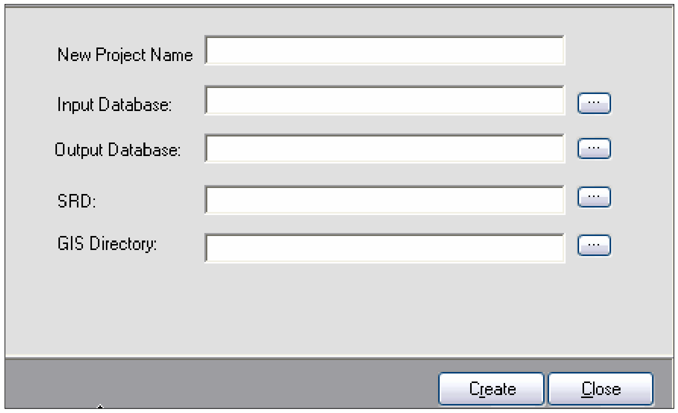



Figure 23. Project Manager to create from existing.

**Create clone of an existing project.** Used to create an exact copy of an existing project in the system. Project cloning allows you to make a duplicate project. Users must select the project to be cloned from a drop-down list of existing projects, type in a name for the new project to be created and specify an existing directory location where the cloned project databases will be created (Figure 24).

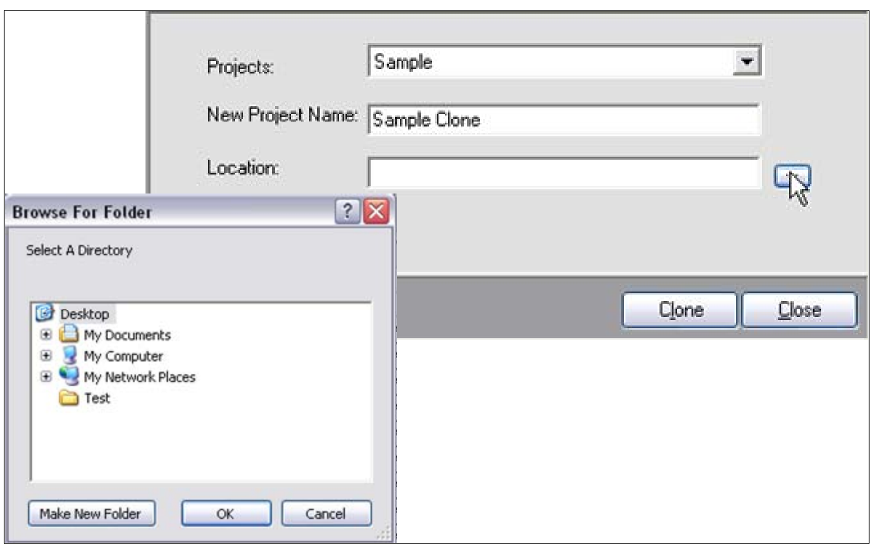
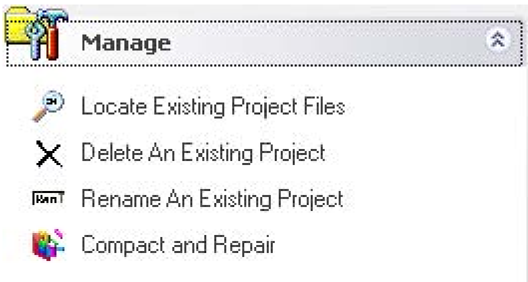



Figure 24. Project Manager to create clone.

### Manage




**Locate existing project files.** Beach-*fx* projects are made up of three separate database files: Input, Output, and Storm Response. If the files for a particular project are moved to a different physical location on the hard drive, Beach-*fx* will be unable to open the project. This feature allows users to locate missing files, enabling the project to be opened by the system.

**Delete an existing project.** Used to delete a project, which is no longer needed, from your system. Users can remove the project from the BEACH *fx* system while leaving the project files (i.e., databases) on their computer, allowing the project to be added back to the system later, or if desired, they can choose to permanently remove the database files from their computer. It is important to note that a project cannot be deleted if it is the currently opened project.


**Rename an existing project.** Used to change the name of any project currently in the system.

**Compact and repair.** As you continue to use Beach-*fx*, the database files can become fragmented and use disk space inefficiently, which will lessen performance over time. This utility will repair and optimize the database files and improve system performance.

# 4 Organization of Data

The key components of a Beach-*fx* project are Profiles, Reaches, Lots, and Damage Elements. These components are related in a hierarchical manner as shown in Figure 25. A project is made up of one or more Profiles (cross-sections of the beach) which characterize the beach morphology in one or more Reaches (contiguous shoreline segments). Each Reach contains one or more Lots (plots of land) and each Lot comprises one or more Damage Elements (something that can incur damage, i.e., house, sidewalk, restaurant, etc.).

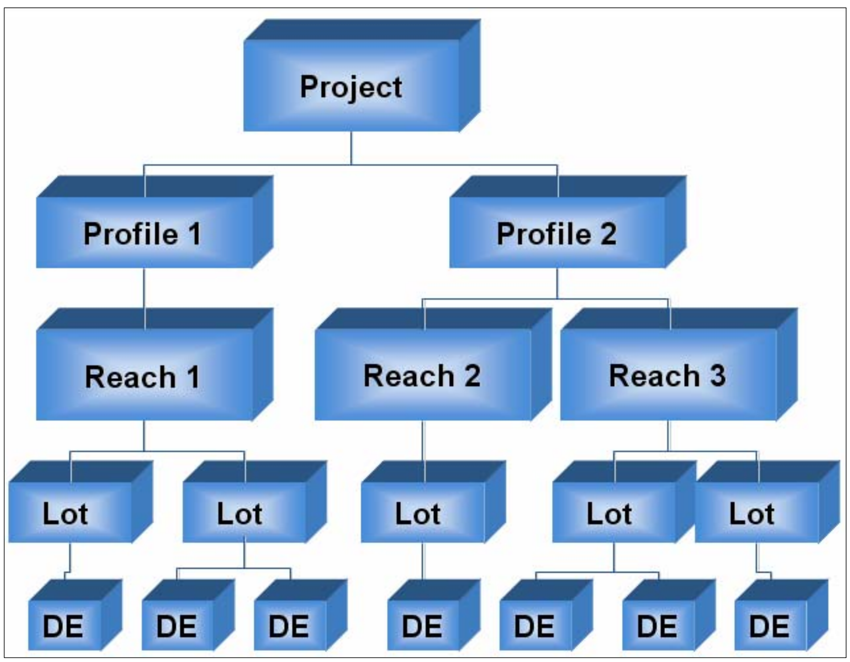



Figure 25. Hierarchial representation of Beach-*fx* data elements.

## Profiles

Coastal process models need to use a detailed distance vs. elevation (X, Z) representation of the beach profile. The amount of data required for such a representation is not needed in an economic-engineering type model such as Beach-*fx* and so a simplified representation for the profile has been adopted. This simplified representation for the profile uses eight key features which include:

1. Dune width.
2. Dune height.
3. Dune slope.
4. Foreshore slope.
5. Upland elevation.
6. Upland width.
7. Berm width.
8. Berm height.

The simplified Beach-*fx* Profile is represented schematically in Figure 26 and the following assumptions apply:

1. A single dune.
2. A single berm (constant elevation).
3. A representative (static) submerged profile.

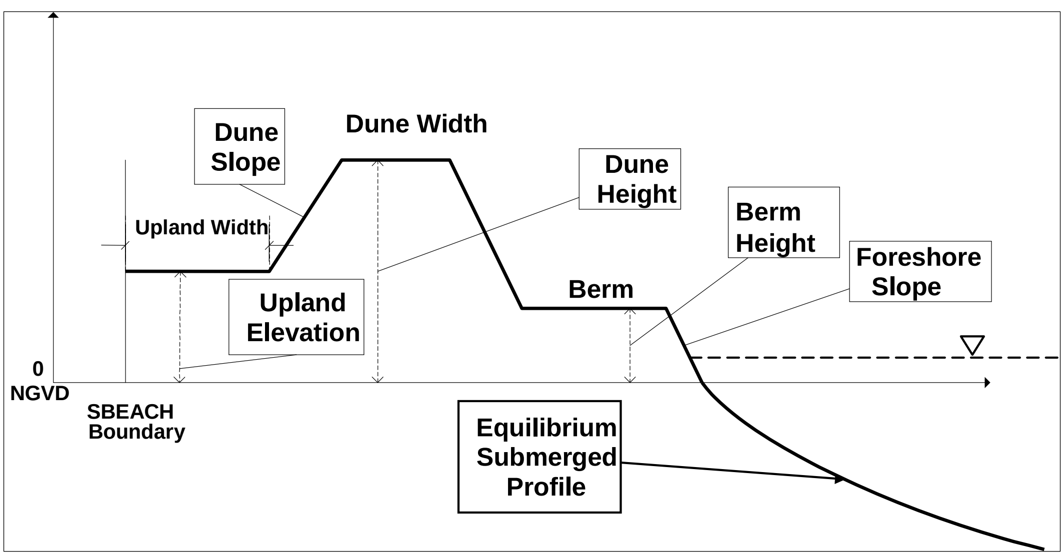



Figure 26. Profile schematic.

## Profile specific data requirements

**Number.** An auto-generated or user-defined integer value greater than zero that can be used to uniquely identify each Profile in the project.

**Description.** User-defined description of the Profile that must match the Profile name used in the SBEACH simulations.

**Default Dune Height (feet).** The height of the dune, measured from datum (elevation 0).

**Default Dune Width (feet).** The width of the top of the dune in feet.

**Default Berm Height (feet).** The height of the berm, measured from datum (elevation 0).

**Default Berm Width (feet).** The width of the top of the berm.

**Dune Slope.** The side slope of the dune. The landward and seaward dune slope is assumed to be constant and equal.

**Foreshore Slope.** The beach slope from the berm to datum.

**Upland Elevation (feet).** Representative ground elevation landward of the dune, measured from datum (elevation 0).

**Depth Of Closure (feet).** Depth below datum to which nourishment material is distributed.

**Detailed Submerged Profile.** A Boolean flag indicating whether the Profile's submerged profile is defined by a set of user-specified X, Z data points or if a representative equilibrium profile is generated instead.

**Submerged Profile Parameter.** A user-specified sediment scale parameter used in calculating the equilibrium profile if the Detailed Submerged Profile Flag is not set. The equilibrium profile is defined by the relationship $depth = Ax^{2/3}$ where A is the empirical sediment scale parameter and $x$ is the distance offshore from the shoreline.

## Reaches

Reaches are contiguous stretches of the shoreline that share a common morphological makeup with a particular Profile. The shoreline of a study can be broken up into any number of Reaches.

Reaches are defined in Beach-*fx* by two straight lines, oriented parallel to the local shoreline, which denote the landward and seaward boundaries of the Reach. The Reach boundary lines are defined by start and end point coordinates, rather than by a set of four points defining a quadrilateral in a clockwise or counterclockwise sequence. This definition (shown in Figure 27) has implications for defining Reach objects in GIS shapefiles and is discussed in more detail in Chapter 9.

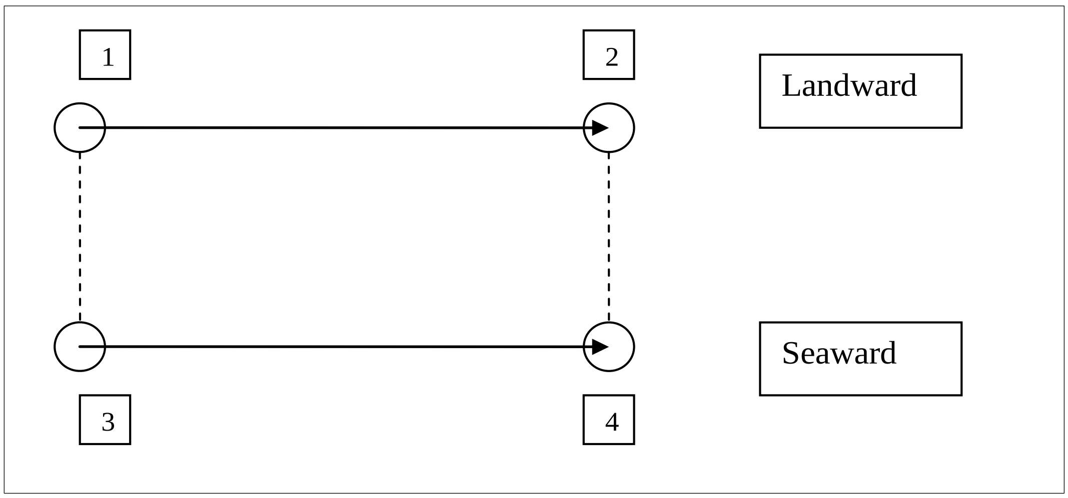



Figure 27. Reach definition.

## Reach specific data requirements

**Number.** An auto-generated or user-defined integer value greater than zero that can be used to uniquely identify each Reach in the project.

**Description.** A user-defined name for the Reach.

**Length (feet).** The shore-parallel distance represented by the Reach.

**Applied Erosion Rate (feet/year).** Feet/year erosion (-) or accretion (+), calibration parameter. The expected rate of shoreline change in the absence of storm events.

**Back-Bay Flooding.** A Boolean flag indicating whether or not back-bay flooding is possible for the Reach.

**Planned Nourishment.** A Boolean flag indicating whether or not the Reach will receive Planned Nourishment.

**Emergency Nourishment.** A Boolean flag indicating whether or not the Reach will receive Emergency Nourishment.

**Upland Width (feet).** The width of the upland area behind the dune.

**Flooding Threshold (feet).** The threshold elevation at which back-bay flooding is initiated.

**Economic Reach Number.** The number of economic Reach, for output purposes only.

**Control Line Offset.** Threshold distance in feet measured from Lot centroid to the seaward toe of the dune at which Lots in the Reach will be marked as condemned prohibiting the rebuilding of Damage Elements in that Lot.

**Start Point Easting.** GIS coordinate of X (easting) of Reach Start Point on Reference Line (Point 1, Figure 27).

**Start Point Northing.** GIS coordinate of Y (northing) of Reach Start Point on Reference Line (Point 1, Figure 27).

**End Point Easting.** GIS coordinate of X (easting) of Reach End Point on Reference Line (Point 2, Figure 27).

**End Point Northing.** GIS coordinate of Y (northing) of Reach End Point on Reference Line (Point 2, Figure 27).

**Shoreward Start Point Easting.** GIS coordinate of X (easting) of Reach Start Point Shoreward (Point , Figure 27).

**Shoreward Start Point Northing.** GIS coordinate of Y (northing) of Reach Start Point Shoreward (Point 3, Figure 27).

**Shoreward End Point Easting.** GIS coordinate of X (easting) of Reach End Point Shoreward (Point 4, Figure 27).

**Shoreward End Point Northing.** GIS coordinate of Y (northing) of Reach End Point Shoreward (Point 4, Figure 27).

**SBEACH Landward Boundary Easting.** GIS coordinate of X (easting) of SBEACH Landward Boundary.

**SBEACH Landward Boundary Northing.** GIS coordinate of Y (northing) of SBEACH Landward Boundary.

**SBEACH Seaward Boundary Easting.** GIS coordinate of X (easting) of SBEACH Seaward Boundary.

**SBEACH Seaward Boundary Northing.** GIS coordinate of Y (northing) of SBEACH Seaward Boundary.

**Berm Width Recovery Factor.** Percent of storm-induced berm width change that is restored due to post-storm recovery processes.

## Lots

Lots are simply an organizational container in the system for Damage Elements. A Lot can be the entire size of the Reach or the size of an actual plot of land in the study area. Lots should be designed in a way that best suits the needs of the study. A Lot is defined by the specification of four points marking each corner of a quadrilateral. An example Lot representation is shown in Figure 28. The points in the Lot must be entered in a clockwise fashion. P1 does not always have to be the upper left corner, but no matter where P1 starts, P2 and all following points must follow clockwise around the Lot.

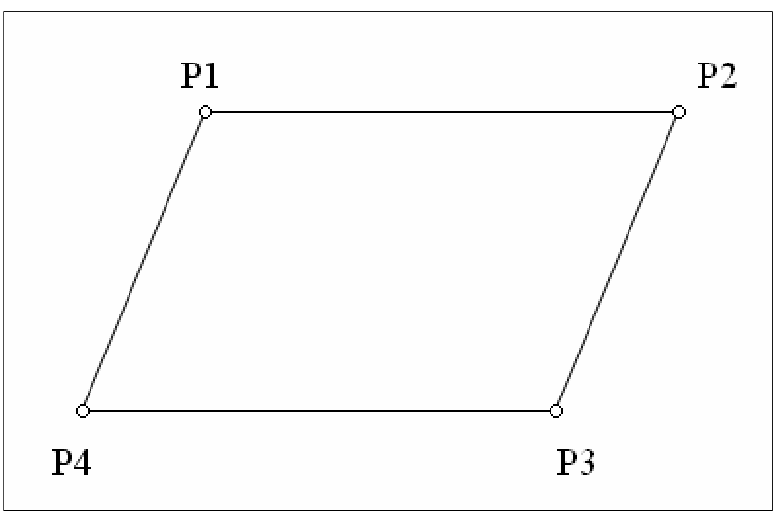



Figure 28. Beach-*fx* Lot representation.

## Lot specific data requirements

**External ID.** An auto-generated or user-defined value that can be used to uniquely identify each Lot in the project.

**Description.** A textual or numeric description of the Lot.

**Type.** A drop-down list containing all valid Lot Types. Currently a Lot can be either Residential or Vacant.

**Armoring Status.** Selected from a drop-down list containing all valid choices: Armorable In The Future, Already Armored, or Not Armorable.

**Erosion (armor failure threshold).** Magnitude of vertical erosion (feet) at the cross-shore location of the armor unit that will cause the armor to fail.

**Flooding (armor failure threshold).** Water-surface elevation at the cross-shore location of the armor unit that will cause the armor to fail.

**Wave Damage (armor failure threshold).** Wave height at the cross-shore location of the armor unit that will cause the armor to fail.

**Distance Trigger (armor construction).** Offset distance (feet) between the seaward edge of the berm and the seawardmost Lot corner that will trigger armor construction on the Lot.

**Length (armor construction).** Length of armor to be constructed on Lot (feet).

**Mobilization Cost (armor construction).** All costs associated with armor construction not included in the Armor Construction Cost Per Foot specification (e.g., engineering and design, equipment rental, backfill material, etc.).

**Cost Per Foot (armor construction).** Estimated cost of armor construction per foot of armor length.

**Mobilization Time (armor construction).** Estimate of time lag between trigger for armor construction and actual initiation of armor construction.

**Time Per Foot (armor construction).** Estimate of the time required for constructing armor, expressed as days per foot of armor length.

**P1 Northing, P1 Easting.** The Northing and Easting coordinates for point P1 (Figure 28).

**P2 Northing, P2 Easting.** The Northing and Easting coordinates for point P2 (Figure 28).

**P3 Northing, P3 Easting.** The Northing and Easting coordinates for point P3 (Figure 28).

**P4 Northing, P4 Easting.** The Northing and Easting coordinates for point P4 (Figure 28).

## Damage Elements

A Damage Element represents any item where damages can be incurred. This could be a house, deck, pool, walkover structure, etc. Damage Elements are members of a Lot and are defined by a single, representative central point (X, Y coordinates).

## Damage Element specific data requirements

**External ID.** An auto-generated or user-defined value that can be used to uniquely identify each Damage Element in the project.

**Description.** A textual description of the Damage Element.

**Type.** Chosen from a drop-down list of all defined Damage Element Types entered into the system (see “Damage Element Types” in Chapter 5).

**Foundation Type.** Chosen from a drop-down list of all defined Foundation Types entered into the system (see “Foundation Types” in Chapter 5).

**Construction Type.** Chosen from a drop-down list of all defined Construction Types entered into the system (see “Construction Types” in Chapter 5).

**Armor Type.** Chosen from a drop-down list of all defined Armor Types entered into the system (see “Armor Types” in Chapter 5).

**Length (feet).** Shore perpendicular distance.

**Width (feet).** Shore parallel distance.

**Number Of Floors.** Number representing the total number of floors in the Damage Element.

**Time To Rebuild (rebuild attributes).** Triangular distribution of rebuild time in days (minimum, most likely, maximum).

**Number Of Rebuilds (rebuild attributes).** Maximum number of times the Damage Element can be rebuilt.

**Representative Point Easting.** The X coordinate of the Damage Element.

**Representative Point Northing.** The Y coordinate of the Damage Element.

**Contents Value.** Triangular distribution of contents value (minimum, most likely, maximum).

**Structure Value.** Triangular distribution of structure value (minimum, most likely, maximum).

**First Floor Elevation.** Triangular distribution of First Floor Elevation (minimum, most likely, maximum).

# 5 Additional Project Related Data

In addition to the application specific/geospatially defined data components in the system (i.e., Profiles, Reaches, Lots, and Damage Elements), there are several other broad systemic data elements which must be initially set up before beginning a Beach-*fx* project. The following sections describe these additional data items.

## Armor Types

All Armor Types found throughout the study area must be entered into the system. This would include Armor Types for both the with- and without-project conditions. A NULL Armor Type is already provided, which is used to signify the absence of armor. If the study area does not involve any type of armor, no additional data entry are required. Armor Type is one of the four Damage Element attributes that are used to identify which damage functions are used for the estimation of storm-induced damages.

## Armor Type specific data requirements

**Code.** A textual identifier for the Armor Type.

**Description.** Textual description of the Armor Type.

**Number.** Auto-generated or user-defined unique number for the Armor Type, used as an external identifier.

## Construction Types

Construction Types are used as distinguishing identifiers amongst Damage Elements, such as Wood or Masonry construction. The Beach-*fx* system is shipped initially with several predefined construction types, but this list can be augmented to meet specific project application needs. Construction Type is one of four Damage Element attributes that are used to identify which damage functions are used for the estimation of storm-induced damages.

## Construction Type specific data requirements

**Code.** A textual identifier for the Construction Type.

**Description.** Textual description of the Construction Type.

**Number.** Auto-generated or user-defined unique number for the Construction Type, used as an external identifier.

## Damage Element Types

Damage Element Types are used as a distinguishing identifier among Damage Elements such as Single or Multi-Family Residential, Commercial, Pool, or Walkway. The system is shipped with several predefined Damage Element Types, but this list can be augmented to meet specific project application needs. Damage Element Type is one of four Damage Element attributes that are used to identify which damage functions are used for the estimation of storm-induced damages.

## Damage Element Type specific data requirements

**Code.** A textual identifier for the Damage Element Type.

**Description.** Textual description of the Damage Element Type.

**Number.** Auto-generated or user-defined unique number for the Damage Element Type, used as an external identifier.

**Linear Element.** A Boolean flag indicating whether the element is considered to be linear such as pools, walkways, tennis courts, etc. Damage Elements defined as Linear Elements should also be marked as auto locate elevation Damage Elements.

**Linear Element Division Length.** Linear Elements can be divided into sections. This parameter specifies the length (feet) of each section for each specific Damage Element Type.

**Auto Locate Elevation.** A Boolean flag indicating whether a Damage Element of a specific type should have its elevation calculated automatically based on the elevation of the parent Profile for that particular location.

**Elevation Offset (feet).** An integer value representing an offset distance to be applied to a Damage Element's elevation if that Damage Element is of that specific type. Used for linear and auto locate elevation Damage Elements.

**Elevation Variability (feet).** This input expresses the uncertainty in the Damage Element elevation attribute specification and is used to establish the minimum and maximum elevation specifications for Damage Element Types marked as linear and auto locate elevation Damage Elements.

**Condemnation Ratio.** The ratio of post-storm structure value divided by initial structure value below, which will result in the structure being marked as condemned, provided that the Damage Element Type is also marked as condemnable. For example, if a condemnation ratio of 0.7 is specified for single-family residential (SFR1), structures and SFR1 Damage Element Types are marked as condemnable structures then, a SFR1 Damage Element will be marked as condemned, if a storm produces damages that exceed 30 percent of the initially specified structure value.

**Condemnable Flag.** A Boolean flag indicating whether Damage Elements of that type are condemnable.

## Foundation Types

All Foundation Types found throughout the study area must be entered into the system. Foundation Types are completely user-defined and typical values would include slab, 8 ft pile, 16 ft pile, etc. A few types have been shipped with the Beach-*fx* system, but this list can be augmented to meet specific project application needs. Each Foundation Type stores its critical erosion amount along with some other identifying elements. Foundation Type is one of four Damage Element attributes that are used to identify which Damage Functions are used to estimate storm-induced damages.

## Foundation Type specific data requirements

**Code.** A textual identifier for the Foundation Type.

**Description.** Textual description of the Foundation Type.

**Number.** Unique number for the Foundation Type, used as an external identifier.

**Critical Erosion Amount (feet).** Vertical erosion that will compromise this Foundation Type.

**Shallow Foundation.** A Boolean flag that indicates whether or not foundations of that type are shallow foundations.

## Emergency Nourishment Alternatives

Emergency Nourishment Alternatives are nourishment events that occur due to a catastrophic event along the shoreline (i.e., the dune width less than a specified trigger width).

## Emergency Nourishment specific data requirements

**Name.** A textual identifier for the Emergency Nourishment Alternative.

**Description.** Textual description.

**Mobilization Cost ($).** Project-level Mobilization Cost, per nourishment action.

## Planned Nourishment Alternatives

Planned Nourishment Alternatives are used to define scheduled nourishment cycles and design templates for each Reach. Nourishment cycles are defined as periodic intervals (e.g., every 3 years), which the Reaches can, if needed, be renourished.

## Planned Nourishment specific data requirements

**Name.** A textual identifier for the Planned Nourishment Alternative.

**Description.** A textual description which typically provides more detail than the name.

**Start Date.** The date the nourishment alternative goes into effect.

**Time Increment (years).** Planned renourishment cycle (i.e., the time between Planned Nourishment events).

**Mobilization Cost ($).** Mobilization Cost, per nourishment.

**Default Borrow To Placement Ratio.** Estimated volume of borrow material required to produce a unit volume of stable fill material on the project beach. This is a Reach-specific attribute and is set at the Reach level.

**Mobilization Threshold.** Minimum number of cubic yards of nourishment material to be placed to justify project mobilization costs.

**Type.** Periodic-Tested. This nourishment implementation type assumes testing on a regular cycle. If at the time of testing, the volumetric nourishment need is less than the mobilization threshold, then testing for project mobilization is not reattempted until the next Planned Nourishment cycle. Periodic-Tested is currently the only nourishment type implemented in Beach-*fx*. Additional nourishment implementation types are planned to be added in future versions.

## Nourishment Blackout Windows

Nourishment Blackout Windows are defined time windows in which beach nourishment/construction activities are not allowed during a certain period each year. These windows are typically driven by environmental considerations such as sea turtle or shore bird nesting periods. However, they can also be driven by local community considerations such as heavy beach usage related to tourist seasons. Beach-*fx* allows the specification of Nourishment Blackout Windows to preclude the placement of beach nourishment during the specified Nourishment Blackout Window time period when a simulation is run.

## Nourishment Blackout Window specific data requirements

**Description.** Textual description for the Nourishment Blackout Window.

**Start Month.** Month the Nourishment Blackout Window starts.

**Start Day.** Day the Nourishment Blackout Window starts.

**End Month.** Month the blackout window ends.

**End Day.** Day the Nourishment Blackout Window ends.

Note that if the End Month is less than the Start Month, then an end-of-year overlap is assumed, e.g., Start Month = 12, End Month = 2 means a

Nourishment Blackout Window from December of the current year to February of the subsequent year.

## Storm Seasons

Storm Seasons are a list of user-defined Storm Seasons appropriate for the study area. The Storm Seasons are used during the simulation when selecting Specific Storms to run. Storm Seasons are to be defined for the entire year, meaning every day of the year should belong to exactly one Storm Season (leap years are not handled). In an attempt to help the user avoid inadvertent season gaps or overlaps, the end month and day of a season are automatically calculated to be 1 day before the next chronological season's start date. If only one season exists, the end month and day will be set to be 1 day before the entered start date for that season, resulting in a single season that spans the entire year. When additional seasons are added, the end dates of existing seasons will adjust accordingly. Users are not allowed to create a new season or modify an existing season to have a start date that is already assigned to another season. The Storm Season specification includes a feature that allows the user to specify overlap days prior to and following the specified season start and end month and day. These overlap days effectively expand the length of the season for purposes of defining the population of storms from which the random selection of storms will occur. For example, if a season has the specified start month and day of 1 September and end month and day of 30 September with previous and next overlap days specified as 31 then the population of storms from which the random sample will be taken includes all storms in the plausible storm suite with assigned dates of occurrence from 1 August (31 days prior to the season start date) through 31 October (31 days after the season end date).

## Storm Season specific data requirements

**Number.** Unique identifier.

**Description.** Textual description for the season (i.e., extra tropical only).

**Start Month.** Month the season starts.

**Start Day.** Day the season starts.

**End Month.** Month the season ends.

**End Day.** Day the season ends.

**Previous Season Overlap (days).** The number of days prior to the specified season start month and day for purposes of identifying the population of storms from which the random sample will be taken.

**Next Season Overlap.** The number of days after the specified season end month and day for purposes of identifying the population of storms from which the random sample will be taken.

**Probability Of Extra Tropical Storm.** Probability of occurrence of extra-tropical storms during that season.

**Probability Of Tropical Storm.** Probability of occurrence of tropical storms during that season.

**Minimum Storm Arrival Time (days).** Minimum inter-arrival time for storms in season.

**Probability Active.** Boolean flag activating or deactivating storms in season.

**Maximum Extra-Tropical Storms In Season.** The maximum number of extra-tropical storms that can occur in the season.

**Maximum Tropical Storms In Season.** The maximum number of tropical storms that can occur in the season.

## Damage Functions

Damage Functions are completely user-definable within Beach-*fx*. A total of six types of Damage Functions are included and specific functions within these types can be designed for use within the model. These types include erosion damage (contents and structure), inundation damage (contents and structure), and wave damage (contents and structure), a complete list of the types and the definitions of the predefined X- and Y-axis can be found in Table 1.

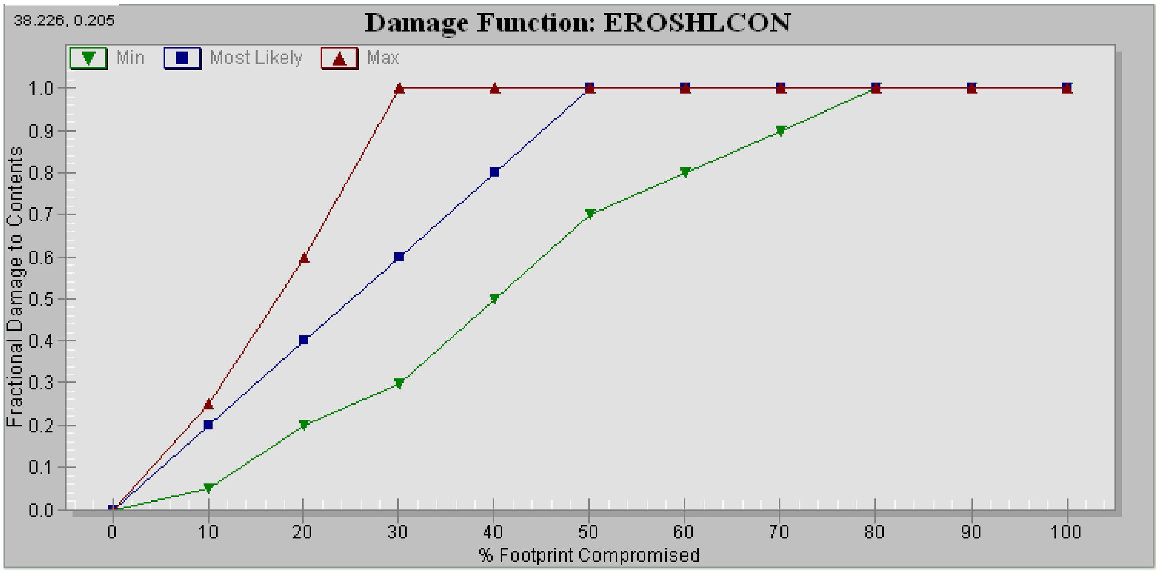
Table 1. Damage Function Types.

| Name | Description | x-Axis | y-Axis |
|---|---|---|---|
| ErosionDamageContents | Erosion Damage to Contents | % Footprint Compromised | Fractional Damage to Contents |
| ErosionDamageStructure | Erosion Damage to Structure | % Footprint Compromised | Fractional Damage to Structure |
| InundationDamageContents | Flood Damage to Contents | Water Depth above 1st Floor | Fractional Damage to Contents |
| InundationDamageStructure | Flood Damage to Structure | Water Depth above 1st Floor | Fractional Damage to Structure |
| WaveDamageContents | Wave Damage to Contents | Water Depth above 1st Floor | Fractional Damage to Contents |
| WaveDamgeStructure | Wave Damage to Structure | Water Depth above 1st Floor | Fractional Damage to Structure |

When the user selects a Damage Function in the *Navigation Pane*, a visual display of that Damage Function is viewable in the *Graphics Pane*. An example of a Damage Function graph can be seen in Figure 29.

The graph displays minimum, most likely, and maximum curves which are defined by values in the function's corresponding data grid. Like other 3-point distributions, the minimum value must be less than or equal to the most likely value, which in turn must be less than or equal to the maximum value. If a value is entered that breaks this rule, a message is displayed warning the user of the invalid entry and they are not permitted to navigate away from that record until the invalid data have been corrected.

Damage Functions are used within Beach-*fx* to estimate storm-induced damages incurred by structures and contents due to three damage driving parameters; erosion, inundation, and wave attack. When multiple damage drivers produce damages to a structure or contents of a structure during a single storm event, Beach-*fx* computes the combined damages according to the following rules to avoid double counting of damages.




| | Fractional Damage to Structure | | |
|---|---|---|---|
| % Footprint Compromised | Min | Most Likely | Max |
| 0 | 0 | 0 | 0 |
| 10 | 0.05 | 0.2 | 0.25 |
| 20 | 0.2 | 0.4 | 0.6 |
| 30 | 0.3 | 0.6 | 1 |
| 40 | 0.5 | 0.8 | 1 |
| 50 | 0.7 | 1 | 1 |
| 60 | 0.8 | 1 | 1 |
| 70 | 0.9 | 1 | 1 |
| 80 | 1 | 1 | 1 |
| 90 | 1 | 1 | 1 |
| 100 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 |

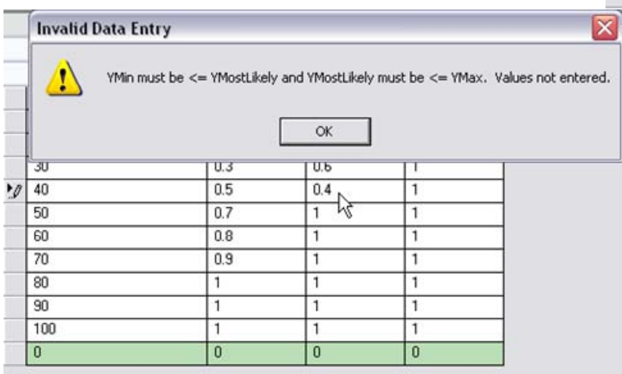



Figure 29. Damage Function graph and data grid.

**Erosion and flood damages.** Combined damages equal the sum of the erosion and flood damages less the product of the erosion and flood damages as follows:

$$\text{Combined Damages} = (\%I + \%E) - (\%I * \%E)$$

where

$\%I$ = fractional loss due to flooding

$\%E$ = fractional loss due to erosion

**Flood and wave damages.** Combined damages equal the sum of the flood and wave damages less the product of the flood and wave damages as follows:

$$\text{Combined Damages} = (\%I + \%W) - (\%I * \%W)$$

where

| | | |
|---|---|---|
| %I | = | fractional loss due to flooding |
| %W | = | fractional loss due to wave attack |

**Wave and erosion damages.** Combined damages equal the sum of the erosion and wave damages less the product of the erosion and wave damages as follows:

$$\text{Combined Damages} = (\%W + \%E) - (\%W * \%E)$$

where

| | | |
|---|---|---|
| %W | = | fractional loss due to wave attack |
| %E | = | fractional loss due to erosion |

**Erosion, flood, and wave damages.** If the Foundation Type is a shallow foundation the combined damages are equal to the maximum damages due to any one of the damage drivers. If the Foundation Type is not a shallow foundation (pile foundation), the combined damages equal the sum of the erosion and the maximum of the flood or wave damages less the product of the erosion and the maximum of the flood or wave damages as follows:

**Shallow foundation**

$$\text{Combined Damages} = \text{Maximum}\ (\%I, \%E, \%W)$$

**Pile foundation**

$$\text{MaxFactor} = \text{Maximum}\ (\%I, \%W)$$

$$\text{Combined Damages} = (\%\text{MaxFactor} + \%E) - (\%\text{MaxFactor} * \%E)$$

## Damage Function Matrix

Once the individual Damage Functions have been defined, the user is required to set up the Damage Function Matrix within Beach-*fx*. The Damage Function matrix is created internally by the application and contains a record for every unique combination of Damage Element Type, Foundation Type, Construction Type, and Armor Type used by the project's Damage Elements crossed with the defined damage types (inundation, erosion, and wave) and the damage components (structure and contents). Users cannot add or delete records from the grid. As shown in Figure 30, users are required to assign a specific Damage Function to each combination record by selecting the desired function from a drop-down list in the grid's first column.

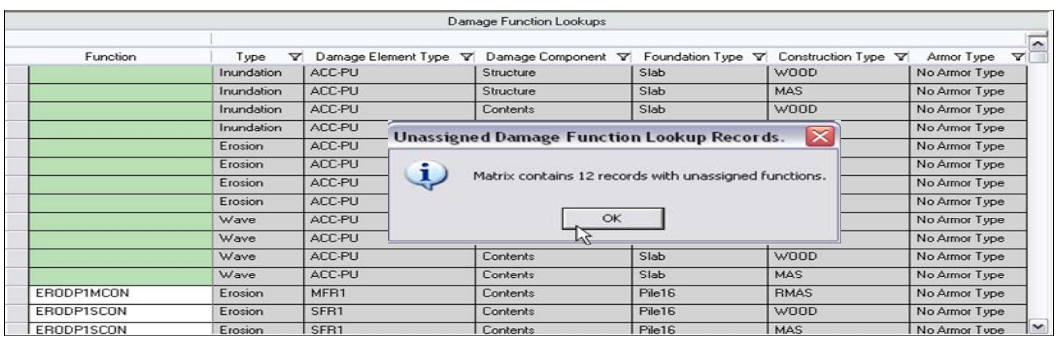



Figure 30. Damage Function Matrix.

In order to ensure all simulations will run properly, the matrix should be fully specified by the user, meaning it should not contain any records for which a function has not been selected. Because the matrix is created from the damage component combinations assigned to the project's Damage Elements, modifying or adding new Damage Elements may result in a previously fully specified matrix containing undefined records. Since the matrix can be quite large, it is important that users be able to easily find and assign functions to undefined combinations. To help with this task, Beach-*fx* will notify users if any such records exist and place those records at the top of the grid (Figure 31).

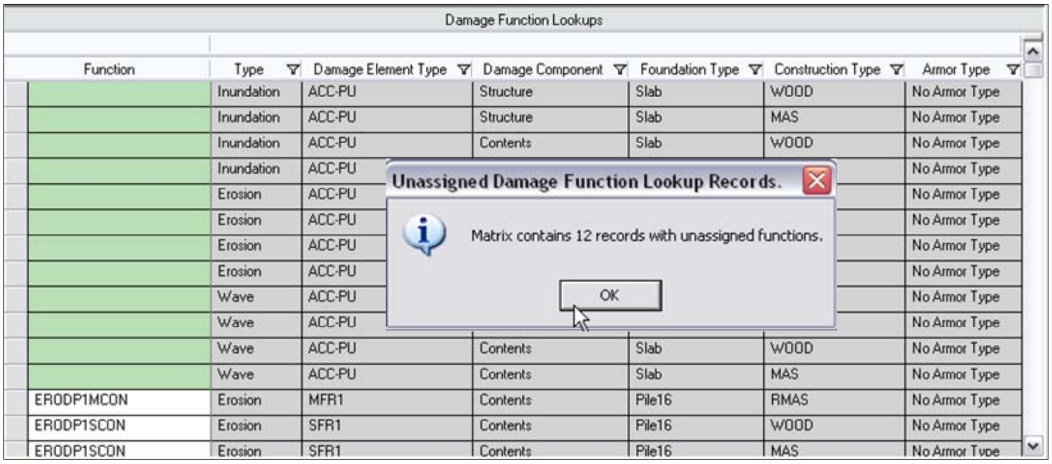



Figure 31. Unassigned matrix records.

## Storms

One or more storms need to be defined within Beach-*fx*. These storms will comprise the plausible storm suite used by the Kernel during simulations. Storms can be either manually entered or imported from an Excel spreadsheet or a comma-separated (CSV) file.

## Storm specific data requirements

**Identifier.** A textual name/description that uniquely identifies the storm. This must be identical to the storm name used in the SBEACH simulations.

**Type.** The type of storm (tropical/extra-tropical).

**Relative Probability.** The relative probability between the storms in the plausible storm suite. For instance a storm with a relative probability of 2 is twice as likely to be selected as a storm with a relative probability of 1.

**Peak Surge Plus Tide.** The elevation of flooding from the back bay.

**Date Of Storm.** The historical date of the storm. This date is used to assign the storm to a defined Storm Season.

## Specific Storms

Once a list of storms has been defined within Beach-*fx*, the user can select Specific Storms to occur during the simulation on a certain date. Regular storm selection during simulations is random. Specifically selecting a storm allows the user to guarantee that an exact storm will occur at the exact chosen date during the simulation.

There are a couple of restrictions on the specification of Specific Storms. Specific Storms must be assigned to occur on a date within the current Scenario's defined time frame. Also, no more than one Specific Storm can be set to occur on a particular date. Attempting to run a simulation with these conditions violated will result in a warning message and cancellation of the simulation (for further information see the "Scenarios" section in Chapter 7).

## Specific Storm data requirements

**Storm.** The unique storm identifier. This is selected from the list of existing storms.

**Date Of Storm.** The exact date the storm should occur during a simulation.

**Active.** A Boolean flag indicating whether or not the specific storm should be counted in the pool of active storms. Only active storms will be considered.

# 6 Reach Level Nourishment

Nourishment Alternatives are defined at a project level, but the assignment of specific nourishment details occurs at the Reach level. If the Reach node in the *Navigation Tree* is expanded, the child nodes Emergency Nourishment, Planned Nourishment, and Reach Planform Rate become visible (Figure 32). Selecting these nodes allows users to enter Reach-specific nourishment data into a corresponding table displayed in the *Data Entry Pane*.

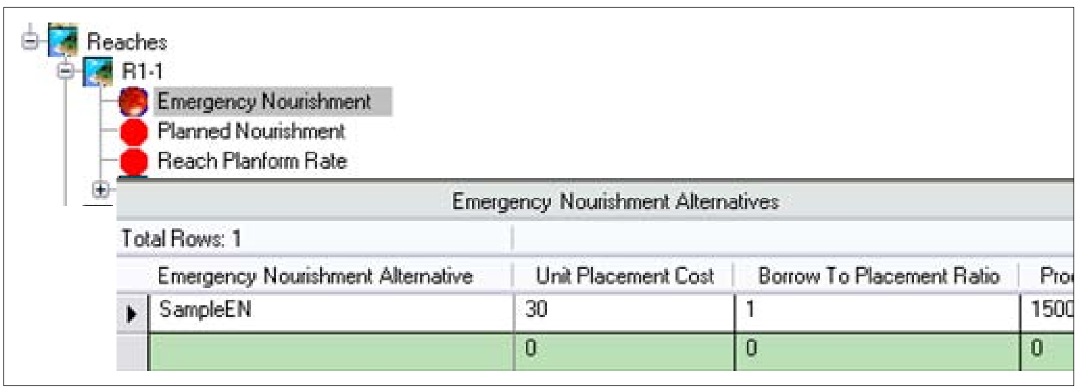



Figure 32. Reach level nourishment.

## Emergency Nourishment

The present implementation of Emergency Nourishment within Beach-*fx* is limited to specification of a nourishment fill density (cubic yard/feet) that acts to increase dune width (at the current dune elevation) at the expense of berm width. However, if the current beach morphology is in a scarping condition, the fill material is first used to restore the berm for the deficit volume represented by the scarping condition. It is possible that the scarping-induced volume deficit may be greater than the specified Emergency Nourishment fill density. In this case, the scarping condition is reduced but not entirely restored and the dune width will remain unchanged.

## Reach level Emergency Nourishment data requirements

**Emergency Nourishment Alternative.** The Emergency Nourishment Alternative to be assigned to the parent Reach. This is selected from a drop-down list containing all Emergency Nourishment Alternatives defined at the project level.

**Unit Placement Cost ($).** The estimated cost of constructing an Emergency Nourishment project expressed as a cost per cubic yard of fill material.

**Borrow To Placement Ratio.** The estimated volume of borrow material required to produce a unit volume of stable fill material on the project beach. This ratio, often referred to as the overfill ratio, accounts for volumetric losses due the sorting and winnowing of fines contained in the fill material.

**Production Rate.** The rate at which fill volume is placed on the beach to construct the Emergency Nourishment project expressed in units of cubic yards per day.

**Mobilization Cost ($).** The Reach-specific costs associated with the nourishment event not included in the Unit Placement Cost attribute.

**Mobilization Time (days).** The estimated time lag between the triggering event and the initiation of Emergency Nourishment construction.

**Priority Order.** An integer value which will determine which Reach receives nourishment first, if more than one Reach is triggered for Emergency Nourishment at the same time. Ordering should begin at 1 and assigned values should be unique.

**Dune Height (feet) (Emergency Nourishment Trigger).** Currently an inactive attribute.

**Dune Width (feet) (Emergency Nourishment Trigger).** A specified dune width that will trigger the first Emergency Nourishment. Subsequent Emergency Nourishments are triggered when the post Emergency Nourishment dune width is further reduced by an amount exceeding the Emergency Nourishment Trigger Adjustment value specified in the Configuration Settings table (see the “Emergency Nourishment configuration settings” section in Chapter 6).

**Berm Width (feet) (Emergency Nourishment Trigger).** Currently an inactive attribute.

**Dune Height (feet) (Emergency Nourishment Template).** Currently an inactive attribute.

**Dune Width (feet) (Emergency Nourishment Template).** Currently an inactive attribute.

**Berm Width (feet) (Emergency Nourishment Template).** Currently an inactive attribute.

After Emergency Nourishment specifications have been made at the Reach level, the full specification can be viewed in the *Navigation Tree* by clicking on the Emergency Nourishment node located beneath the Nourishment Alternatives For All Reaches. The data table will display all Emergency Nourishment Alternatives for each Reach that has the Emergency Nourishment flag selected (Figure 33).

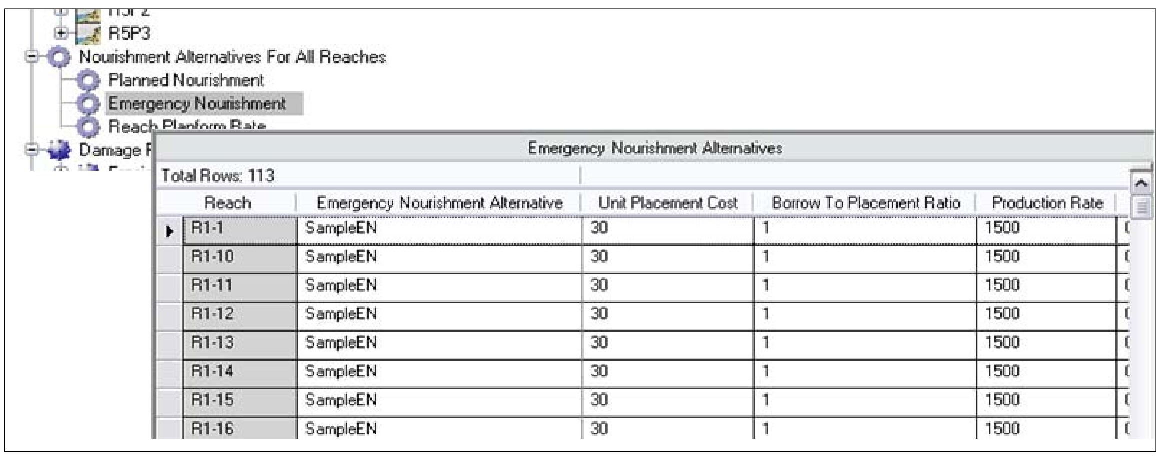



Figure 33. Emergency Nourishment Alternatives for all Reaches.

## Emergency Nourishment configuration settings

A number of Emergency Nourishment attributes are specified in the Configuration Settings table. The Configuration Settings table can be displayed by clicking on the **Configuration Settings** node located beneath the Project Information node of the *Navigation Tree* (Figure 34). The related settings are as follows:

Configuration Settings

Total Rows: 35

| | Tag | Description | Value |
|---|---|---|---|
| | DataCheckMaximumLotSize | Maximum Lot Size, in acres, for data check flag (lots above this | 5 |
| | DataCheckMinimumLotSize | Minimum Lot Size, in acres, for data check flag (lots below this | 0.1 |
| | DepthOfClosureMultiplier | Multiplier on entered depth of closure for use in morphology ch | 0.1 |
| | EmergencyNourishmentFillDensitySpecification | Cubic Yards/Foot Fill Density For Emergency Nourishment | 4 |
| | EmergencyNourishmentMobilizationTimeThreshold | Time Window after end of last Emergency Nourishment at whic | 20 |
| | EmergencyNourishmentScheduledNourishmentBlackoutWind | Multiplier on reach emegency nourishment mobilization to set b | 2 |
| ▶ | EmergencyNourishmentTriggerAdjustment | Emergency Nourishment Trigger Adjustment > 0 set trigger gre | 1 |
| | LookupBermWidthExceptionReportTolerance | Value in feet for flagging exception report on lookups for berm | 20 |
| | LookupDuneHeightExceptionReportTolerance | Value in feet for flagging exception report on lookups for dune | 20 |

Figure 34. Emergency Nourishment configuration settings.

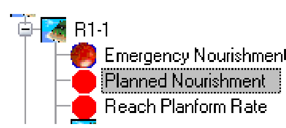
**Emergency Nourishment Fill Density Specification.** The Emergency Nourishment Fill Density Specification (cubic yard/feet).

**Emergency Nourishment Trigger Adjustment.** The additional dune width loss (feet) that will trigger a subsequent Emergency Nourishment action after construction of the initial Emergency Nourishment project.

**Emergency Nourishment Mobilization Time Threshold.** A time window specification (days) beyond which any subsequent Emergency Nourishment will incur another project level mobilization cost.

**Emergency Nourishment Scheduled Nourishment Blackout Window Multiplier.** A multiplier (factor) that is applied to the Emergency Nourishment mobilization time and checked against the start date of any Planned Nourishment activity. If a Planned Nourishment activity is scheduled to begin on or before the calculated date then the Emergency Nourishment activity is canceled. This is a necessary scheduling issue that will prohibit an overlapping of Emergency Nourishment and Planned Nourishment activities. Users can enter appropriate values for these settings by editing the value column. These specifications are global to the project and do not vary by Reach.

## Planned Nourishment




The present implementation of Planned Nourishment within Beach-*fx* (periodic-tested) involves nourishment trigger specifications expressed as a percent of specified nourishment template values along with a target renourishment interval, start date, mobilization threshold, and mobilization costs. On the start date, the required nourishment volume is

estimated for all Reaches in which at least one of the threshold trigger specifications is satisfied. If the required nourishment volume exceeds the mobilization threshold volume, then a Planned Nourishment activity is scheduled. When nourishment occurs, all Reaches are restored to the specified nourishment template regardless of the nourishment threshold triggers. Renourishments are processed in a similar manner at the specified renourishment interval. If the mobilization threshold volume is not exceeded, nourishment does not take place. The next Planned Nourishment check will occur at the end of the renourishment interval.

## Reach level Planned Nourishment data requirements

**Planned Nourishment Alternative.** The Planned Nourishment Alternative to be assigned to the Reach. This is selected from a list of Planned Nourishment Alternatives the user defined at the project level.

**Unit Placement Cost ($).** The estimated cost of constructing a Planned Nourishment project expressed as a cost per cubic yard of fill material.

**Borrow To Placement Ratio.** The estimated volume of borrow material required to produce a unit volume of stable fill material on the project beach. This ratio is often referred to as the overfill ratio and accounts for volumetric losses due the sorting and winnowing of fines contained in the fill material.

**Production Rate (cubic yards/day).** The rate at which fill volume is placed on the beach to construct the Planned Nourishment project expressed in units of cubic yards per day.

**Processing Order.** An integer value indicating the order Reaches will be processed for nourishment if multiple Reaches are set to receive nourishment at the same time.

**Mobilization Cost ($).** The costs associated with Reach-specific mobilization costs related to the nourishment event and not included in the Unit Placement Cost attribute.

**Dune Height (feet) (nourishment trigger).** Fractional amount of template dune height that denotes requirement for renourishment.

**Dune Width (feet) (nourishment trigger).** Fractional amount of template dune width that denotes requirement for renourishment.

**Berm Width (feet) (nourishment trigger).** Fractional amount of template berm width that denotes requirement for renourishment.

**Dune Height (feet) (template).** The post-construction dune height.

**Dune Width (feet) (template).** The post-construction dune width.

**Berm Width (feet) (template).** The post-construction berm width.

After Planned Nourishment specifications have been made at the Reach level, the full specification (all Reaches) can be viewed by clicking on the **Planned Nourishment** node located beneath the Nourishment Alternatives For All Reaches node in the *Navigation Tree* (Figure 35). The data table will display all Planned Nourishment Alternatives for each Reach that has the Planned Nourishment flag selected.

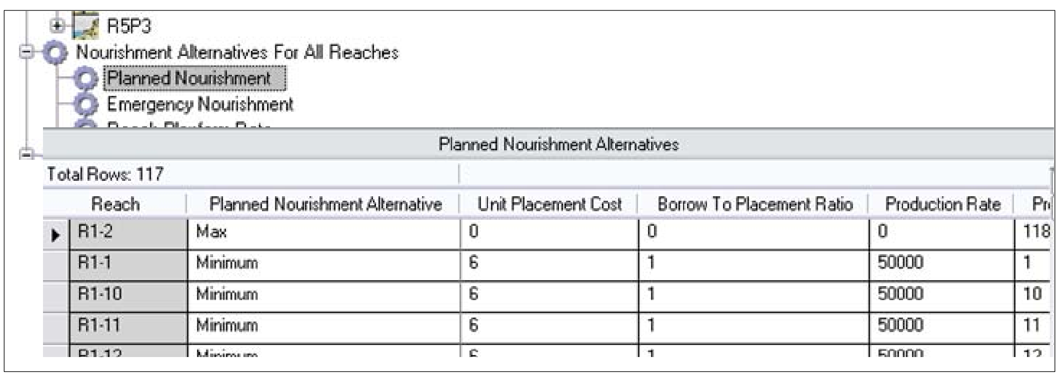



Figure 35. Planned Nourishment Alternatives for all Reaches.

## Reach Planform Rate

The final step in completing the Planned Nourishment specification is to provide Reach-level estimates of the project-induced shoreline rate of change.


Beach nourishment is the placement of relatively large quantities of high quality fill material (sand with grain size characteristics similar to the natural beach) on a beach to advance the shoreline seaward and to provide elevation by way of a dune feature adequate for the protection of the

upland area. The beach nourishment project represents a planform perturbation on the natural shoreline and is characterized as seaward displacement of the shoreline. This disequilibrium in the planform induces sediment flows that over time reduce the extent of the planform disequilibrium, resulting in the beach nourishment project approaching equilibrium through alongshore dispersion of the fill material. This spreading-out (or dispersion) process of beach nourishment material from the placement area is captured within Beach-*fx* by way of specification of project-induced shoreline change rates or the Reach Planform Berm Width Change Rate. This rate is typically estimated using a shoreline change model such as GENESIS. The rate of shoreline change due to the dispersion process varies with location along the placement area, with greater rates of change near the lateral ends of the project and lesser rates of change near the center of the placement area. Consequently, the Reach planform rates of change are specified at the Beach-*fx* Reach level. Further, because the rate of fill material dispersion changes with maturation (age) of the nourishment project, this input is further specified to vary by nourishment cycle. Based on numerical model simulations, the user should estimate the average rate of project-induced shoreline rate of change by nourishment cycle for each Reach within and adjacent to the nourishment placement area.

## Reach Planform Rate data requirements

**Planned Nourishment Alternative.** The Planned Nourishment Alternative specified for the Reach. This is selected from a list of Planned Nourishment Alternatives the user defined at the project level.

**Nourishment Cycle.** An integer representing the nourishment cycle. Each construction event represents the beginning of a new nourishment cycle.

**Berm Width Change Rate.** The rate of project-induced shoreline change (feet/year) resulting from the placement of beach nourishment material. Estimates of this input are typically obtained from shoreline change model results. This input is expected to vary spatially by Reach and over time by nourishment cycle.

After completing the specification of the Reach planform rate for individual Reaches, the full specification can be viewed by clicking on the *Navigation Tree* node **Reach Planform Rate** located beneath the

Nourishment Alternatives For All Reaches node in the *Navigation Tree* (Figure 36).

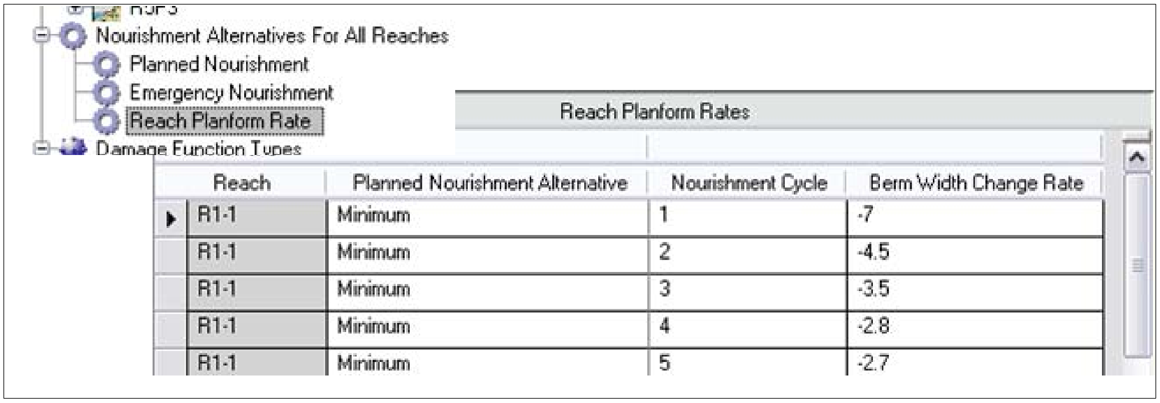



Figure 36. Reach Planform Rate for all Reaches.

# 7 Scenarios and Simulation

## Scenarios

Beach-*fx* simulation parameters are defined using a Scenario. Individual simulation Scenarios are created by the user to specify a desired set of simulation parameters. Selecting the Scenarios header node in the *Navigation Tree* displays a data grid containing all existing Scenarios. Users can create a new Scenario by either typing information into the blank row at the bottom of the grid or by right-clicking anywhere on the Scenario branch of the tree and selecting **Add -> New Scenario** from the corresponding context menu. Once a Scenario is created, selecting its node in the *Navigation Tree* displays the Scenario control panel (Figure 37). From this screen, users can edit the Scenario specific data requirements. Any changes made to the Scenario through this screen are not saved to the database until the user selects the **Save** button at the bottom of the panel. This means if the user runs a simulation before saving, none of the entered changes will be taken into account by the simulation engine.

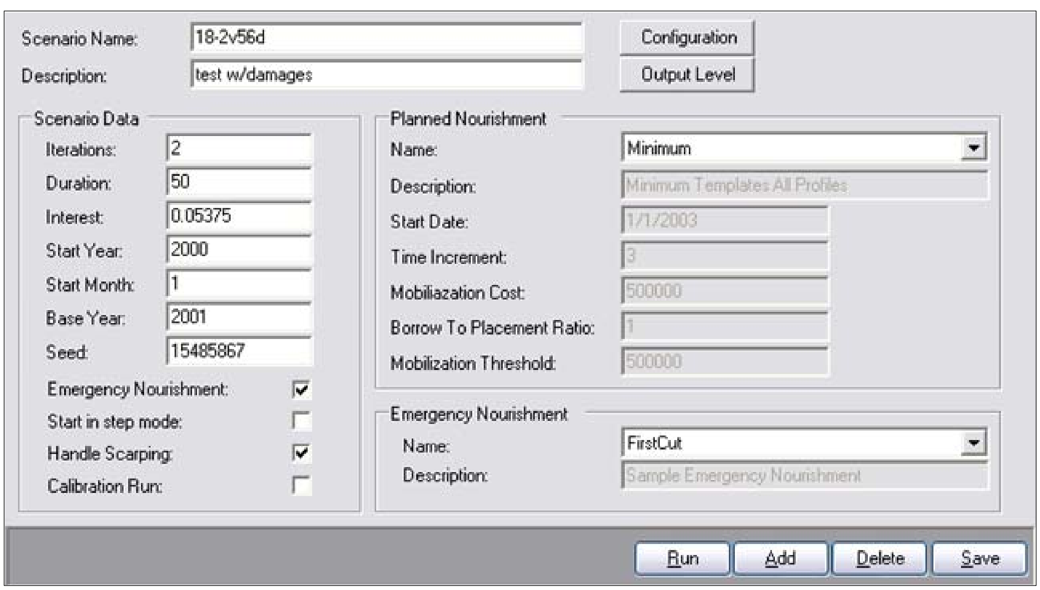



Figure 37. Scenario control panel.

**Scenario specific data requirements:**

**Name.** Scenario names should be unique.

**Description.** A textual description of the Scenario.

**Start Year.** The simulation start year for the Scenario.

**Start Month.** The simulation start month of the Scenario.

**Base Year.** Reference year for present value calculations. This must be greater than the simulation Start Year.

**Emergency Nourishment Flag.** A Boolean flag indicating whether Emergency Nourishment will be applied for the Scenario during the simulation.

**Planned Nourishment Alternative.** Defined Planned Nourishment Alternative that will be applied for this Scenario simulation.

**Emergency Nourishment Alternative.** Defined Emergency Nourishment Alternative that will be applied for this Scenario simulation.

**Step Flag.** A Boolean flag indicating if the simulation should enter Step Mode to allow the user to single step the simulation.

**Iterations.** The number of life-cycles or simulations to be performed.

**Duration.** The number of years spanned in each iteration/life cycle. The life-cycle duration should equal the number of years between the simulation start year and the base year (the year the Planned Nourishment Alternative is in place and producing benefits) plus the economic analysis study period (typically 50 years for the U.S. Army Corps of Engineers Hurricane and Storm Damage Reduction projects).

**Interest.** The interest rate for present value calculations.

**Seed.** Integer seed value for the random number generator which is used to generate the storm sequence during a simulation. It is recommended that the seed be specified as a large prime number. It is important to understand that the declared seed value for successive simulations must be identical in order for those simulations to experience the same

sequence of storm events. If different Scenarios are to be inter-compared (e.g., with and without project) the seed value must be the same to ensure the same random sequence of storms is encountered in each Scenario.

**Scarping.** A Boolean flag indicating whether the Scenario should include detailed treatment of dune scarping (recommended).

**Calibration.** If checked, simulation will involve only morphology change calculations and preclude all economic related calculations.

There are some data restrictions when configuring a Scenario to run. If the entered base year is less than or equal to the start year, an error message will be displayed when the user attempts to save the Scenario (Figure 38).


Figure 38. Scenario configuration error.

As mentioned in the “Specific Storms” section in Chapter 5, active Specific Storms are assigned a unique date of occurrence and that date must fall within the Scenario’s specified time frame. The time frame is the start date plus the entered duration. If these requirements are not met, an error message will be displayed and the Scenario will not be run. In Figure 39, there are two active Specific Storms with a date of 01/05/2007. Attempting to run a Scenario in this situation would result in an error message being displayed as shown in Figure 39. The user could either change the date of one of the storms or simply uncheck the **Active** flag to remove one or both of the storms from the pool of storms set to occur during the simulation.


Figure 39. Overlapping Specific Storms.

## Running a simulation

Selecting a specific Scenario from the Navigation Pane will provide access to a **Run** button in the Data Entry Pane. Clicking on this button will launch the simulation engine screen using the settings for that Scenario. The simulation engine itself, shown in Figure 40, allows for visualizations to be generated during the simulation run. An example of the simulation engine, along with a description of its components can be found next.

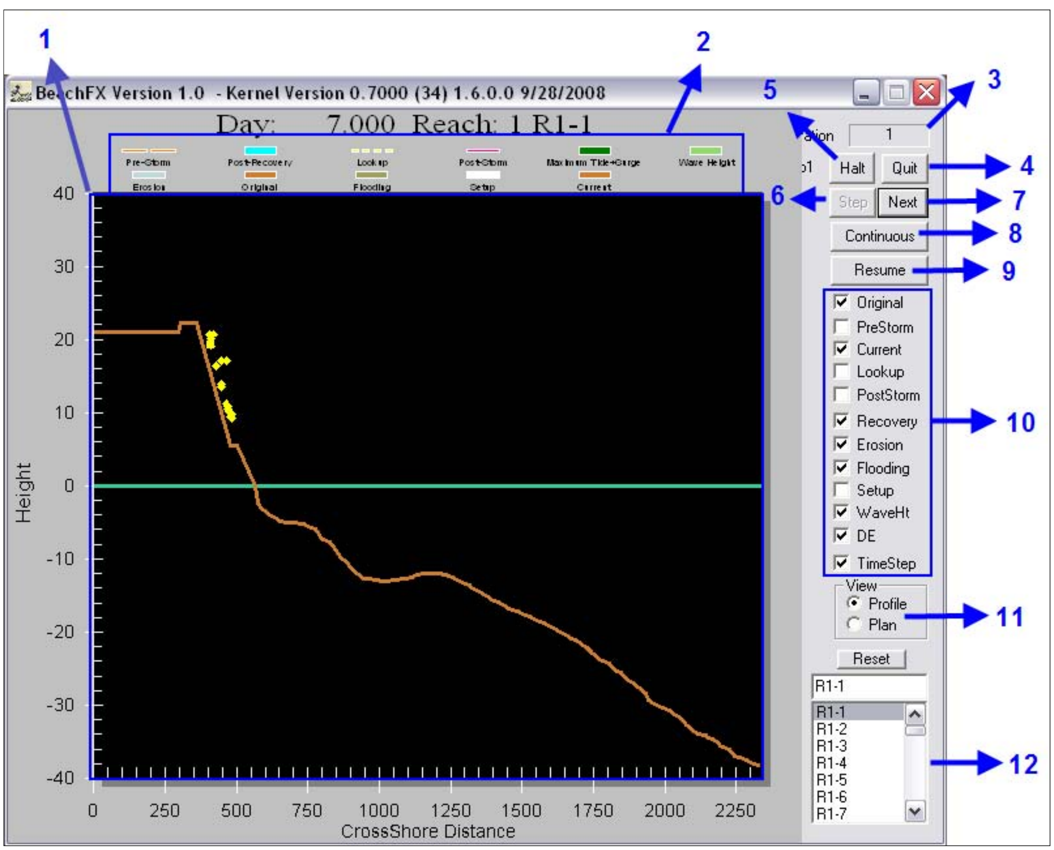



Figure 40. Beach-*fx* simulation interface.

1. Main display for the simulation routines. Displays any of the items found in item #10.

2. Graph legend.

3. Current iteration the simulation is running.

4. Stops the simulation.

5. Stops the simulation and closes the Kernel at the end of the current iteration.

6. Places the simulation in Step mode, which gives the user a visual display of what is happening in the system and allows them to step through each time-step.

7. Moves to the next time-step.

8. Continuously steps through each time-step while still displaying the visual representation in the graph.

9. Resumes the simulation in normal mode. This will not provide a visual display to the user, which means it runs much faster than step mode.

10. A list of the different items that can be graphed while in step mode. Each item can be turned on and off by checking or un-checking the box next to the item.

11. Flips between *Profile* and *Plan View*. *Plan View* is a top down view of the Reach.

12. List of Reaches in the system. Clicking on a Reach will display that Reach in the graph while in step mode.

# 8 Outputs

## Viewing outputs

Reports and graphs are available to view the raw data output by the simulation engine in a user friendly manner. It is also possible to open and view the generated data files directly from Beach-*fx* by using the File Viewer. Additionally, depending on the output files generated, it is possible to view a two-dimensional (2D) animation of the coastal morphology evolution over the simulation's timeline by Reach and iteration. Once a simulation has been executed on the project, its output is available to be viewed.

The reports, graphs, and file viewer forms all allow the viewing of data on a Scenario specific basis. The desired Scenario is selected from the drop-down list. Only Scenarios that have been run and have data will be available for selection. There is also the option to remove all records for that Scenario from the output database. Choosing this option will result in deletion of all the previously generated data files from the computer. However, the Scenario itself is not deleted; it is as if the Scenario has never been run.

## Reports

The Beach-fx Report Builder, as shown in Figure 41, can be launched by choosing **Results -> Reports** from the application's main menu. The Report Builder is used to set up which report to run and to set various options for that report. Multiple reports can be viewed simultaneously, allowing for comparison between the reports. Reports have the ability to be printed or exported into multiple file formats, including Adobe Acrobat, Microsoft® Excel, Microsoft® Word, and Rich Text Format.

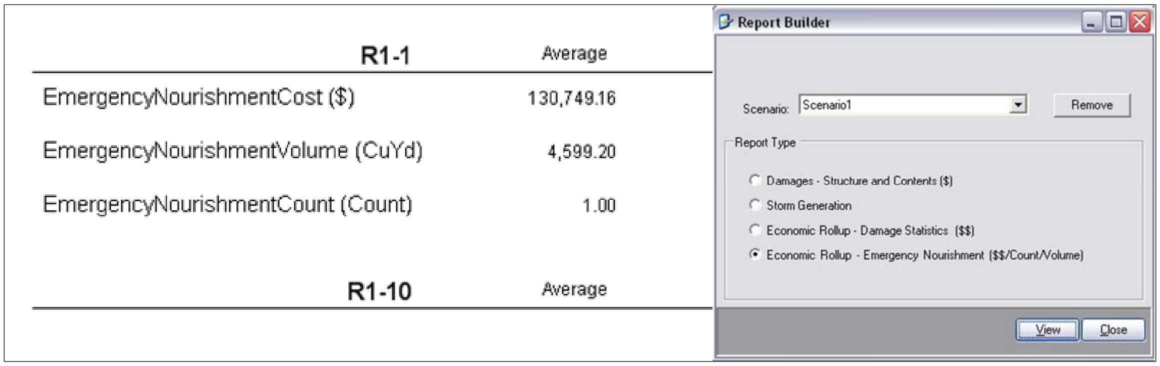



Figure 41. Beach-*fx* report builder.

## Graphs

Beach-*fx* also contains a graphs builder, as shown in Figure 42, which can be launched by choosing **Graphs** under the *Results* main menu. Once a graph has been generated, the entire look of the graph can be altered by the user. Right-clicking the mouse while the cursor is over the graph will bring up a customization menu with a wide range of features for altering the view of the graph. Graphs can be printed or exported to a number of different image formats, including Metafile, BMP, JPG, and PNG. There is also the ability to export the data from the graph to a text file.

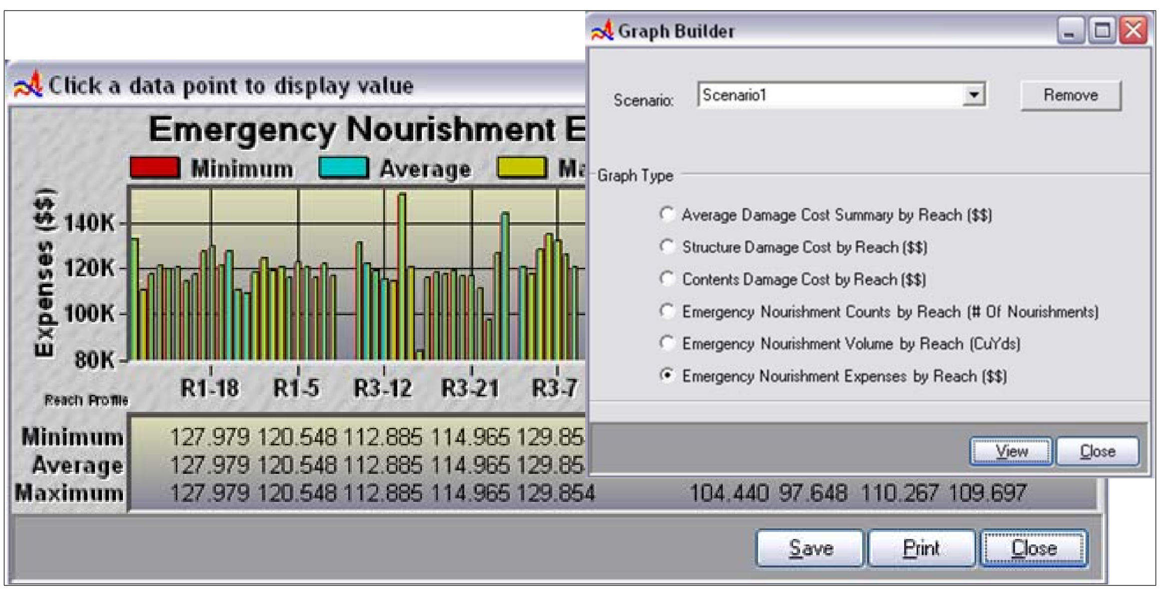



Figure 42. Beach-*fx* graph builder.

## Data files

When a Scenario is run, the generated files are placed in the current project's root directory (the folder where the IDB is located). Although users can browse to appropriate directory and open files they wish to view, it may be more convenient to use the Beach-*fx* File Viewer, which is listed under the application's Results Menu. Like the Report and Graph Builder screens, the File Viewer has a drop-down of all Scenarios that have been run and have associated result data. Data from different Scenarios can be viewed simultaneously by changing the selected Scenario in this list. Select the file to be viewed from the file list and click the **View** button to open it.

The radio buttons in the **Filter Data Files By Category** section limits the files that are listed in the file drop-down box at the bottom of the File Viewer form to those associated with the selected category. If the **All** option is selected, all files from all categories will be listed. The right-most image in Figure 43 displays an example listing of all files generated for Scenario 1. The files are categorized by the kind of data they contain. The first 10 files are considered to be more general in nature and contain data relating to the project and/or the specific Scenario run. The other categories are as follows: Damage, Storm, Nourishment, Reach, Lot, and Damage Element. To filter by a specific category, users can select the appropriate filter option. For example, selecting the **Reach** filter option only lists the files containing Reach related data (Figure 44). A general description of all generated output files is included as Appendix B.

Open output files should be closed before attempting to run the Scenario again. Output files that are left open cannot be edited by the Kernel during the simulation run. In order to prevent the unexpected termination of a simulation because files are locked, users will receive a warning if they attempt to run a Scenario when one or more of that Scenario's output files are open, and the simulation will be cancelled.

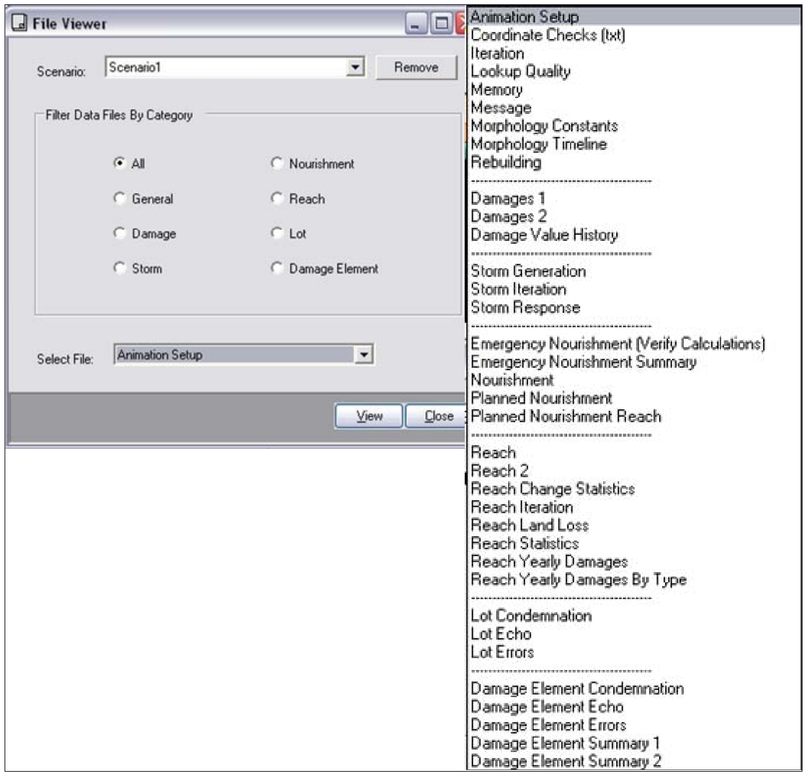



Figure 43. File Viewer.

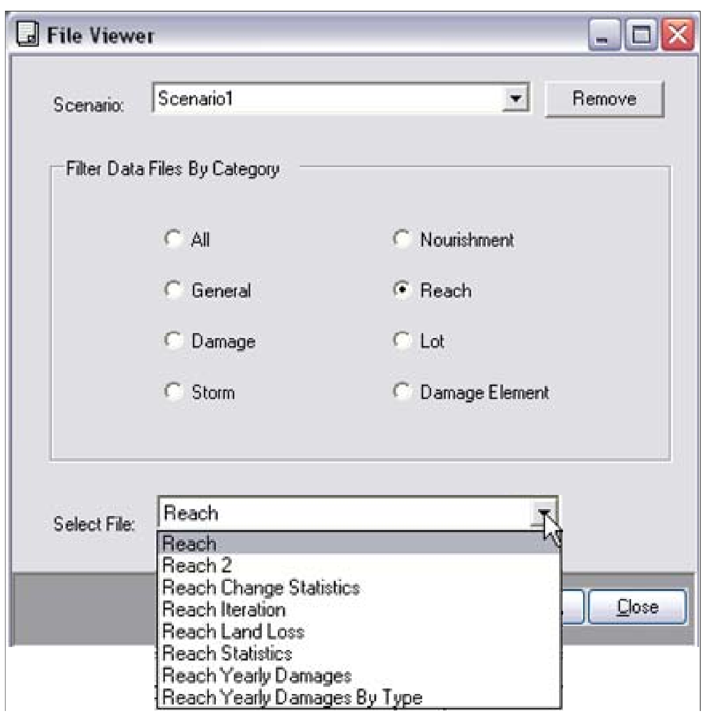



Figure 44. File Viewer, filter for Reach data.

## Animation

If a Scenario has been run with the Output Option **Morphology Timeline** selected, then a 2D animation of the project's morphology evolution can be viewed for that Scenario. The animation form is opened by selecting the **Animation** option under the Results menu.

The animation form is shown in Figure 45. To set up the animation, the Scenario, Reach, and iteration must be specified from the pull-down list. Once these items are selected, the graph panes are loaded, but the animation will not begin until the **Run** button is selected. The form contains three embedded views, each offering a different perspective. The top panel is the Profile View which provides a cross-section view of the Reach's profile. The profile is drawn multiple times to display the initial profile, the current profile, and the profile as it will be affected by the next two morphology changing events. For example, at the occurrence of a storm event, the profiles illustrated are the Pre-storm, Post-storm, and Post-recovery profiles. The *5 Year* and *Total Life Cycle* views both display the Reach from an aerial perspective (Plan View). The *Total Life Cycle* displays the Reach's coastal morphology for the entire simulation life cycle. In Figure 45, the total life cycle spans 50 years. This value is the duration of the Scenario which was run. The *5 Year* view displays the same information as the *Total Life Cycle* view, but in a scrolling, 5-year time frame. Because it is displaying 5 years of data, as opposed to the entire life cycle, the *5 Year* view appears to be magnified and it is easier to see the detailed effects of individual events as they occur.

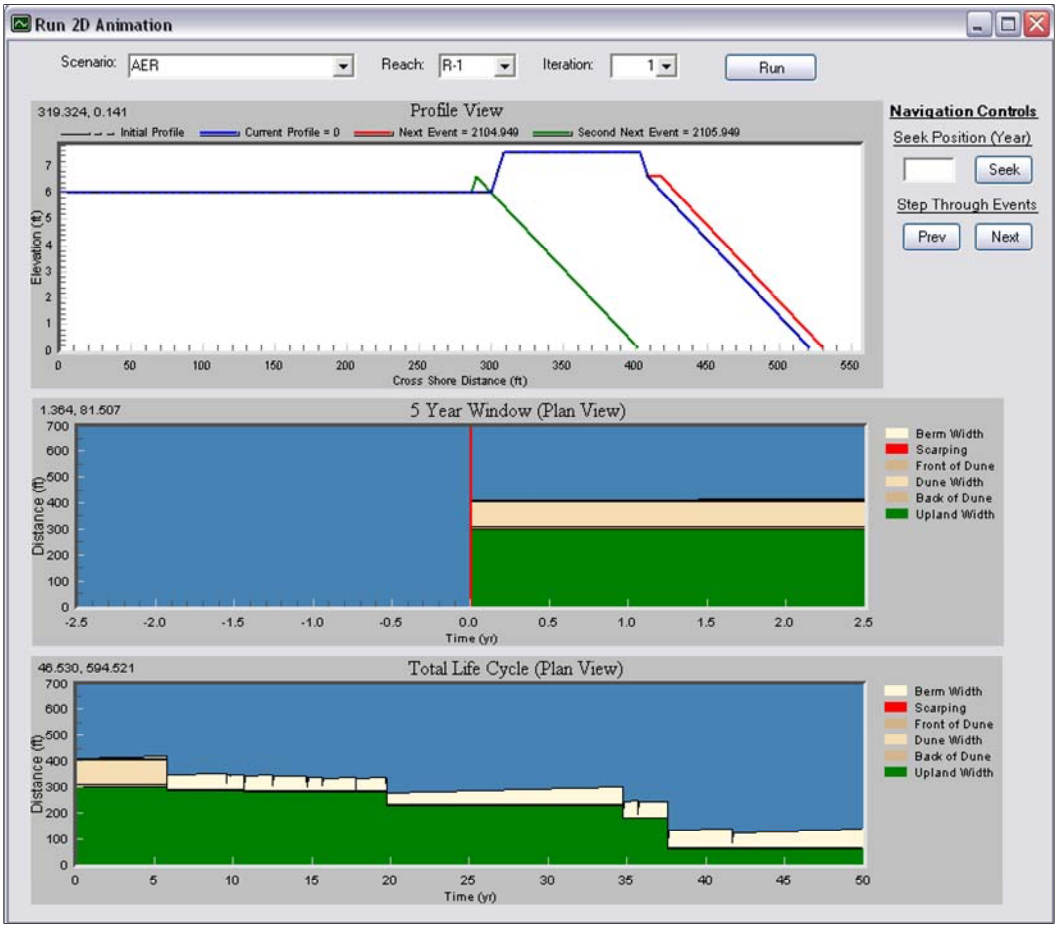



Figure 45. Beach-*fx* animation.

If the user clicks on the **Run** button (which toggles as the **Stop** button), the animation will begin and run continuously—looping repeatedly through the life cycle—until the user clicks **Stop**. Clicking the **Stop** button will halt the animation at its current time, meaning it will not reset. However, the user can instantly jump to a particular time (year) by entering an integer or decimal value into the **Seek Position** textbox and clicking the **Seek** button. The user also has the ability to step through events one by one (backward or forward) simply by clicking on the **Prev** or **Next** buttons located on the side panel.

The animation form provides users with a visual representation of Reach-level coastal morphology evolution for all iterations. Detailed event information, including the event type and related storm, is contained in the file <*ScenarioName*>_MorphologyTimeLine.csv and can be viewed in combination with the animation for a more comprehensive analysis (Figure 46).

AER_MorphologyTimeLine.csv

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Iteration | Time | Reach | UplandWidth | DuneHeight | DuneWidth | BermWidth | EventType | |
| 2 | 1 | 0 | R-1 | 300 | 7.5 | 95 | 0 | INIT | NA |
| 3 | 1 | 2104.949 | R-1 | 300 | 7.5 | 95 | 9.579 | PreStorm | WCe_19641004_H2 |
| 4 | 1 | 2105.949 | R-1 | 285.896 | 6.6 | 0 | 0 | PostStorm | WCe_19641004_H2 |
| 5 | 1 | 2125.949 | R-1 | 285.896 | 6.6 | 0 | 56.29 | REC | WCe_19641004_H2 |
| 6 | 1 | 3490.871 | R-1 | 285.896 | 6.6 | 0 | 62.516 | PreStorm | WCe_18870727_H2 |
| 7 | 1 | 3491.871 | R-1 | 285.896 | 6.6 | 0 | 26.594 | PostStorm | WCe_18870727_H2 |
| 8 | 1 | 3511.871 | R-1 | 285.896 | 6.6 | 0 | 59.02 | REC | WCe_18870727_H2 |
| 9 | 1 | 3550.336 | R-1 | 285.896 | 6.6 | 0 | 59.212 | PreStorm | WCe_18890923_M3 |
| 10 | 1 | 3551.336 | R-1 | 285.896 | 6.6 | 0 | 59.212 | PostStorm | WCe_18890923_M3 |
| 11 | 1 | 3571.336 | R-1 | 285.896 | 6.6 | 0 | 59.307 | REC | WCe_18890923_M3 |
| 12 | 1 | 3576.393 | R-1 | 285.896 | 6.6 | 0 | 59.307 | PreStorm | WCe_19161018_H3 |
| 13 | 1 | 3577.393 | R-1 | 285.896 | 6.6 | 0 | 53.96 | PostStorm | WCe_19161018_H3 |
| 14 | 1 | 3597.393 | R-1 | 285.896 | 6.6 | 0 | 58.868 | REC | WCe_19161018_H3 |
| 15 | 1 | 3900.959 | R-1 | 285.896 | 6.6 | 0 | 60.273 | PreStorm | WCe_19750923_H1 |
| 16 | 1 | 3901.959 | R-1 | 280.443 | 6.6 | 0 | 0 | PostStorm | WCe_19750923_H1 |
| 17 | 1 | 3921.959 | R-1 | 280.443 | 6.6 | 0 | 59.169 | REC | WCe_19750923_H1 |
| 18 | 1 | 4569.466 | R-1 | 280.443 | 6.6 | 0 | 62.107 | PreStorm | WCe_18960707_M1 |
| 19 | 1 | 4570.466 | R-1 | 280.443 | 6.6 | 0 | 19.346 | PostStorm | WCe_18960707_M1 |
| 20 | 1 | 4590.466 | R-1 | 280.443 | 6.6 | 0 | 57.927 | REC | WCe_18960707_M1 |
| 21 | 1 | 5354.477 | R-1 | 280.443 | 6.6 | 0 | 61.407 | PreStorm | WCe_19330802_L2 |

Figure 46. Morphology timeline output file.

## Output Options and Configuration Settings

Users can set the Configuration Settings and Output Levels for the Scenario by selecting the appropriate button (Figure 47). The Output Options grid lists all the output data files that can be generated by the simulation engine during a simulation. Users can select what files they would like to be generated by checking/unchecking the Output Flag field in the table. If a user creates a new empty project using the database templates provided with the application, the table defined in the IDB should already contain all valid Output Options used by the simulation engine. It is important to note that only the Output Flag and Sort Order fields can be edited in the data grid. The user can use the Sort Order field to place those files where the output status is most frequently modified at the top of the table. Like the Output Options, the Configuration Setting options are used by the Kernel during Beach-*fx* simulations. They are predefined in the IDB.

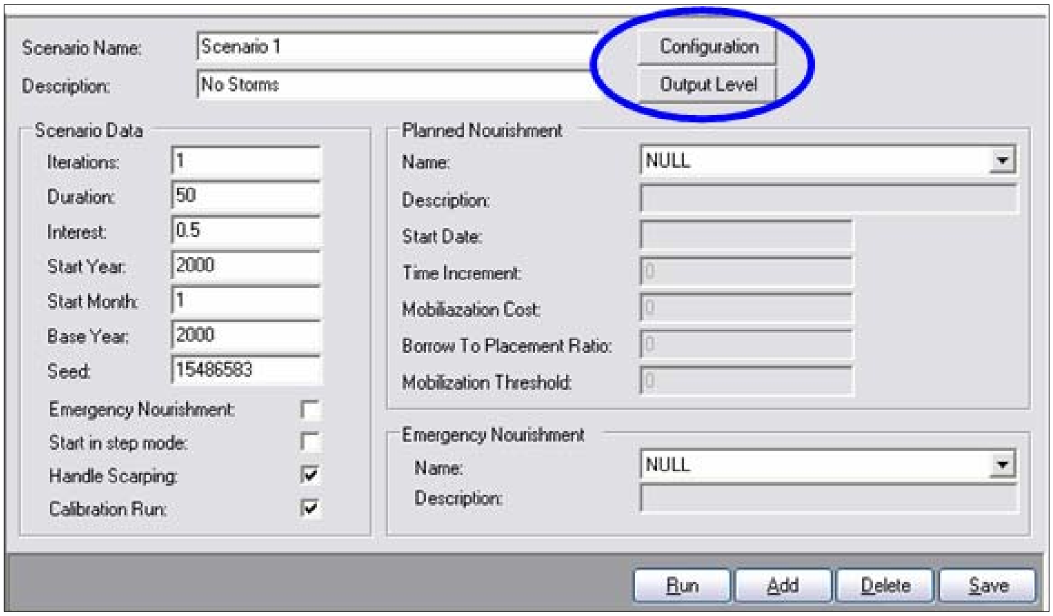


Figure 47. Output Options and Configuration Setting.

### Output Options specific data requirements

**Output flag.** A Boolean flag indicating whether a specific output file should be generated by the Kernel when the simulation is run.

**Sort order.** User-specified integer value, to change placement of rows in the table.

Note: The remaining fields are for informational purposes only and are not modifiable by the user.

**Tag.** A condensed string tag associated with each specific output data file.

**Description.** A detailed textual description of what data the output file will contain.

**Generation.** Frequency at which the associated output is generated.

**File suffix.** Files generated by the Kernel have the naming convention of <ScenarioName> <FileSuffix>. The file suffix includes the portion of the name that uniquely identifies it as well as the associated file extension type.

#### Configuration Setting specific data requirements

**Generation.** A description of when the file is generated during the simulation.

**Tag.** The textual tag associated with the Configuration Setting.

**Description.** A textual description of the purpose for the setting.

**Value.** The value to use for the setting. Note this is the only field editable by the user.

## Data checking

Due to the volume of project data that must be specified, it can be difficult to determine whether a project has all the necessary data entered, and whether that data is considered to be valid. Some data errors become apparent when attempting to run a simulation because they are severe enough that they result in the termination of the Beach-*fx* Kernel. Other data errors simply result in bad or nonsensical output.

Beach-*fx* provides a couple of data checking routines to help users evaluate their project data. Users should be aware that these routines do not check everything. They are still being modified, and additional enhancements will likely be added in the future.

## Checking shoreline structure and IDB data

Users can check shoreline structure data at any time by running the check data routine, accessed by right-clicking the root project node in the *Navigation Tree* (Figure 48). Selecting this option will open a modal form with a scrolling text pane. Various data checks are performed, such as checking that Profiles, Reaches, Lots, and Damage Elements exist and that each object belongs to a valid parent object. Additional checking is completed on each of the object's attributes to make sure that required attributes have been entered, and that the values are within specified ranges. Errors and warnings are displayed as issues are identified. Each message includes the object name, the attribute checked, and the invalid value. A final tally of errors and or warnings uncovered is calculated at the end.

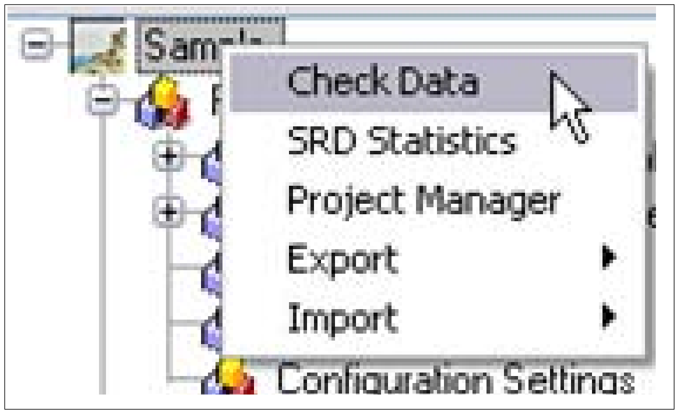



Figure 48. Beach-*fx* check data routine.

Figure 49 shows sample results for a project. In this example, an error message is listed notifying the user that the Damage Function Matrix has not been fully specified. Based on the final totals displayed at the bottom, this is the only error that was detected.

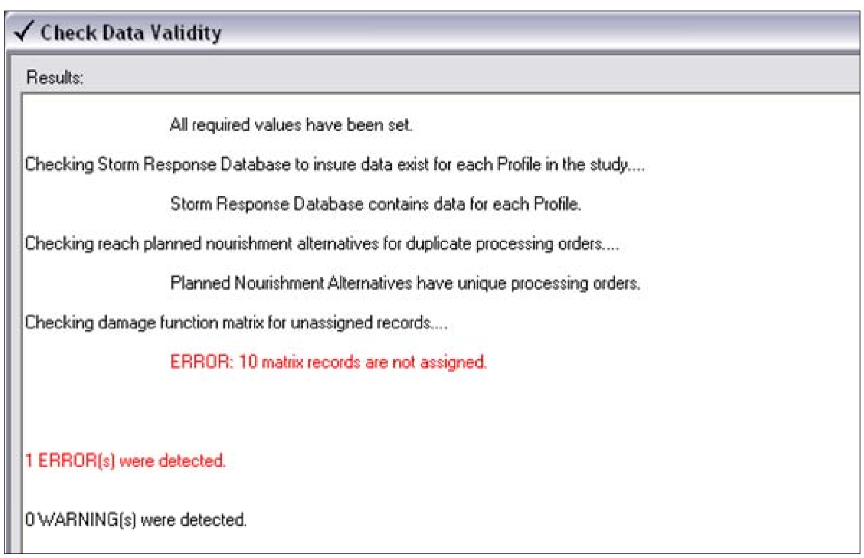



Figure 49. Results panel for check data routine.

## Storm response database statistics

After the Storm Response Database has been populated with SBEACH data, users have the ability to generate and display a handful of summary statistics for each Profile. This is accomplished by right-clicking on the project node in the *Navigation Tree* and selecting the **SRD Statistics** option from the menu.

Figure 50 shows sample output. The SRD statistics routine will ensure that every Profile in the project has corresponding data in the SRD. It also allows users to check that every combination of Profile, berm width, dune width, dune height, and storm is present before attempting to run a Scenario.

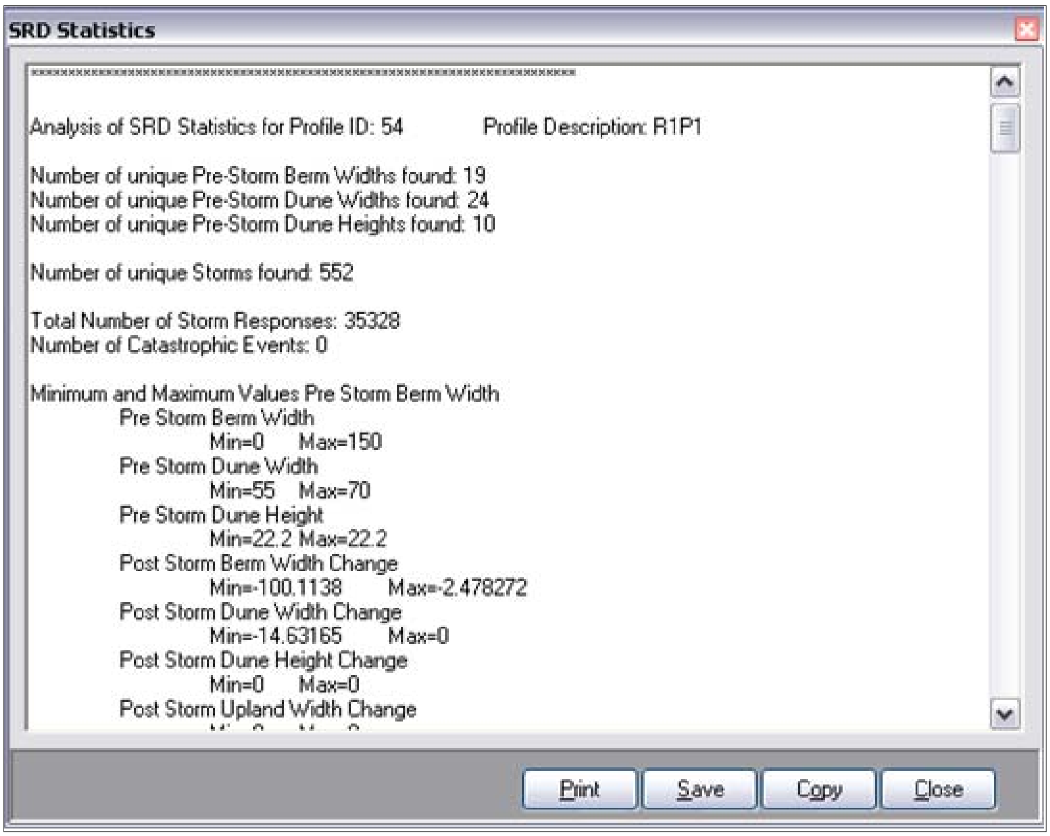



Figure 50. SRD statistics.

# 9 Exporting and Importing Data

Import and export methods have been implemented to assist users in moving data into and out of Beach-*fx*. The **Import** and **Export** submenus are found under the application's **File** menu header. The type of import or export (CSV, Microsoft® Excel, or Shapefile) is listed in parenthesis after the object description.

## Exports

Beach-*fx* data can be exported for individual objects or for the entire project by either right-clicking on the appropriate object node in the *Navigation Tree* or by selecting **File-> Export-> <*Export Type*>** from the main application menu. Because the menu displayed when the user right-clicks on an object node in the *Navigation Tree* is context specific (meaning it can be specific to the individual object selected), there is some functionality available to users through right-clicking that cannot be achieved through the main menu. Separate CSV export methods exist for all Beach-*fx* objects. Additionally, it is also possible to export Reach, Lot, and Damage Element data as shapefiles.

When exporting data, users are allowed to specify the destination directory, but not the individual file names. The default destination is a directory named ObjectExports_<*mm_dd_yyyy*>, created in the same directory as the current project. Individual data files are placed within this subdirectory. If the entire project is exported, the default destination is two levels down from the project home in ObjectExports_<*mm_dd_yyyy*>\<*Current Project Name*>. Users should be aware that if the default directory is selected for the same type of object export more than once on the same day, any existing data files will be overwritten.

Some of the export dialogs group related objects. For instance, instead of having a separate dialog for each of the four Damage Element Variables (Armor Types, Construction Types, Foundation Types, and Damage Element Types), there is a menu item to export Project Information which brings up the following Export Project Information form.

The Export Project Information form displayed in Figure 51 allows users to export any or all of the individual components located under the Project Information header node in the application's *Navigation Tree*. In this example, the user right-clicked on the **Damage Element Variables** header node and selected the **Export** option. Because of this, all four of the Damage Element variable types are already selected. If the user would have selected the **Export** option by right-clicking on the **Armor Types** node, e.g., only that option would have been selected by default when the export form displayed. There is also a **Select All** checkbox which allows users to check/uncheck all components.

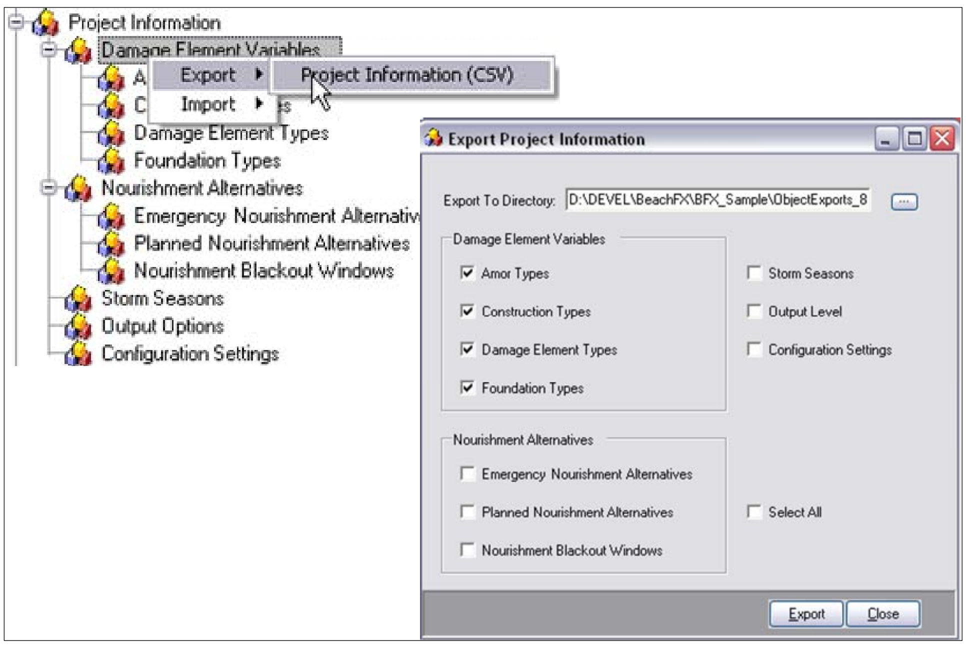


Figure 51. Export Project Information.

For objects that have a hierarchical relationship (Shoreline Structures and Damage Functions), it is possible to selectively export data related to a specific object by right-clicking on the object's node and selecting the appropriate export method. For example, if a user right-clicks on Reach R2-2, the following options would be available as shown in Figure 52.

Selecting **All Shoreline Structures** (CSV) would be the same as selecting **File -> Export -> Shoreline Structures** (CSV) from the main menu. However, if **Object Data** (CSV) is selected, the user is allowed to export only those records that are related to Reach R2-2. Figure 53 shows

a comparison of the export forms displayed when the user selects either option.

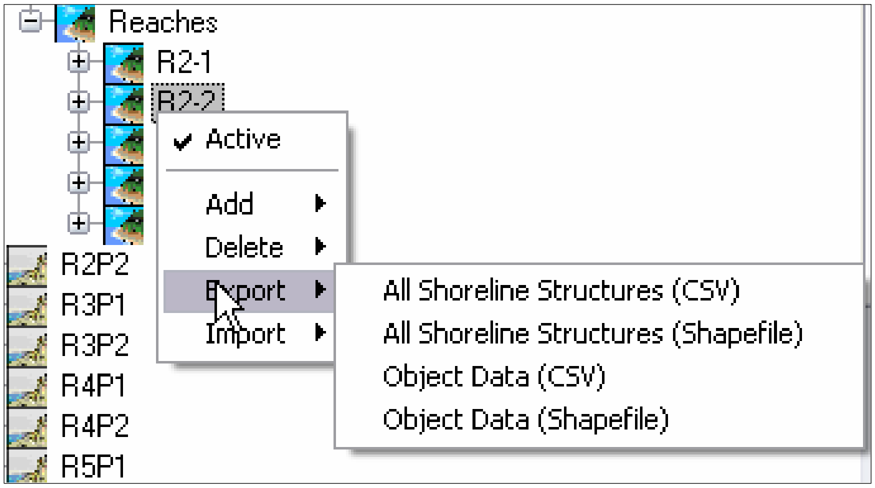



Figure 52. Export options for Reach R2-2.

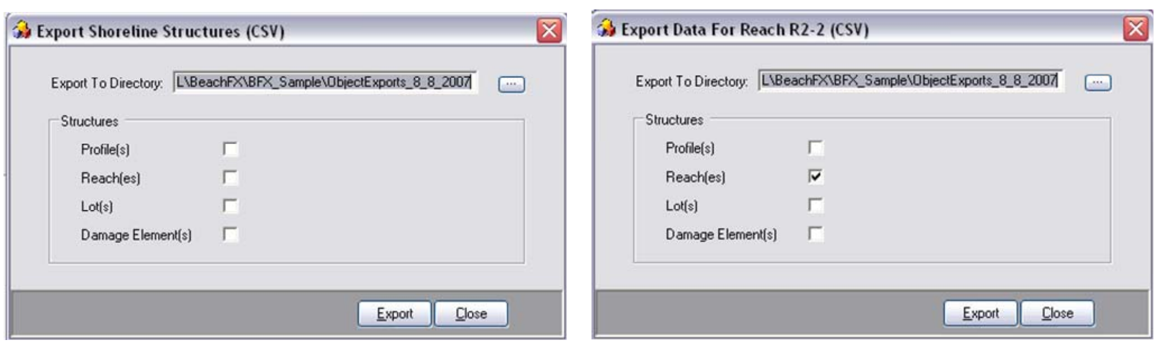



Figure 53. Exporting shorelines structure data, all (left) vs. specific (right).

The form on the left allows for the exportation of all Profiles, Reaches, Lots, and/or Damage Elements depending on the options selected by the user. The form on the right allows for the exportation of all shoreline structures connected to Reach R2-2 through the established hierarchical relationships. By default only the Reaches checkbox is selected. However, if the user selects other checkboxes, all Profile, Reach, Lot, and Damage Element records associated with R2-2 would be exported, resulting in four CSV files named ProfileDataForReach_R2-2_<mm/dd/yyyy>, ReachDataForReach_R2-2_<mm/dd/yyyy>, LotDataForReach_R2-2_<mm/dd/yyyy>, and DEDataForReach_R2-2_<mm/dd/yyyy>. Similar functionality is available for exporting Damage Functions.

## Exporting an entire project

Users can export all data within a Beach-*fx* project by selecting **File -> Export -> Project (CSV)** or by right-clicking on the project's root node in the *Navigation Tree* and selecting **Export -> Project (CSV)**, as shown in Figure 54. The Beach-*fx* project exportation method generates a separate CSV file for each type of data structure. It also generates an INI file, called ProjectImport_<*Current Project Name*>.ini, in the same directory as the generated CSV files. This INI file contains information specifying what kind of object data each CSV file contains as well as each file's location on the user's computer. It is used by the Beach-*fx* project import process so the user does not always have to manually select a template file for each Beach-*fx* component.

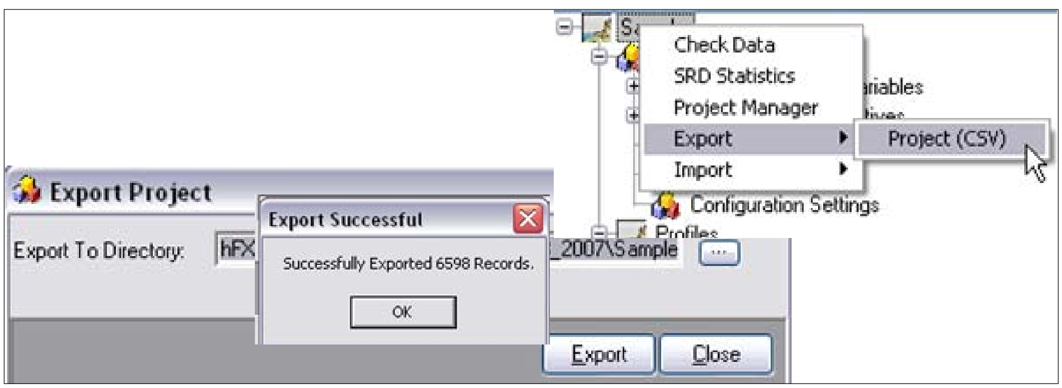



Figure 54. Exporting Beach-*fx* project.

## Exporting shoreline structures as shapefiles

It is possible to export Reaches, Lots, and Damage Elements to separate GIS shapefiles. Profiles are not included because they lack spatial information. The same functionality available when exporting to CSV files is provided when exporting to shapefiles, meaning users can export all structures in a project or a specific branch of the shoreline structure hierarchical tree. The default destination for the files is ObjectExports_<*mm_dd_yyyy*>\Shapefiles.

The shapefiles created through exportation also contain attribute tables constructed and populated in the defined format required for importing the structures back into Beach-*fx*. The attribute table formats for each type of object are listed in Appendix A. For Beach-*fx* projects containing data but lacking GIS components, exporting the structures as shapefiles

provides a way to generate files to be used for the *Map View* in the Beach-*fx* graphics pane. GIS shapefiles are a collection of separate files, each bearing the same name but having a different file extension. The three extensions the user will see for Beach-*fx* generated shapefiles are .shp, .shx, and .dbf. The naming conventions for exported shapefiles follow the same conventions as the CSV exports, with the current date being appended to the name following the structure type. Users can rename these files however they wish, but all related files must be renamed the same. For example, to rename the exported output from AllReaches_08_06_2007 to just Reaches, the renamed files should be Reaches.shp, Reaches.shx, and Reaches.dbf.

## Imports

Aside from the manual method for entering data into the system, import methods have been developed to ease the population of large amounts of data into Beach-*fx*. Depending on the type of data being imported, the user has several different import options. For instance, shoreline structures (Profile, Reaches, Lots, and Damage Elements) can be imported from Microsoft® Excel or CSV templates. Additionally, Reaches, Lots, and Damage Elements can also be imported from shapefile templates, which is discussed in detail in Chapter 10.

Users also have the capability to import SBEACH results from pre-generated SBEACH .dat files. It is necessary to import SBEACH data for every project after the Storms and Profiles have been defined as this routine sets up the storm response linkages between specific storms and the Profiles.

Beach-*fx* import routines are available to users by either right-clicking on the desired object type node in the *Navigation Tree* to bring up that node's associated context menu or by selecting **File -> Import -> <*object_type*>** from the application's main menu. Once selected, an import dialog for the object will open allowing the users to locate and select the appropriate import template(s) (CSV, Excel, shapefile). On nearly all of the import forms, there is a checkbox that specifies whether or not a log file should be generated during the import process. This checkbox is checked by default and the generated log will be placed in the current project's root directory under an *ImportLogs* subdirectory. The log files are named to reflect the type or types of objects for which they contain information. An integer representation of the system date and time is

appended after the text portion of the file name to uniquely identify the log and prevent it from being overwritten the next time the user invokes the same type of import.

There are also two radio buttons on the import form designating the type of import to be performed: **Append** and **Load/Reload**. The **Append** option, which is the default, specifies that new object records will be inserted into the appropriate database table, but no updates or deletions will be performed on objects already existing in the database. Any records found in the template file for objects that already exist in the database will be ignored and a warning message will be written to the log file. The **Load/Reload** method has been designed to fully replace/update object data. If the user selects this option, all objects are deleted from the corresponding table(s) before any records are read from the template file. Thus, if a user selects **Load/Reload** when importing Profiles, all child Reach, Lot, and Damage Element records will be removed. Due to the severe consequences of selecting this option accidentally, users are prompted for confirmation each time this option is chosen (Figure 55).

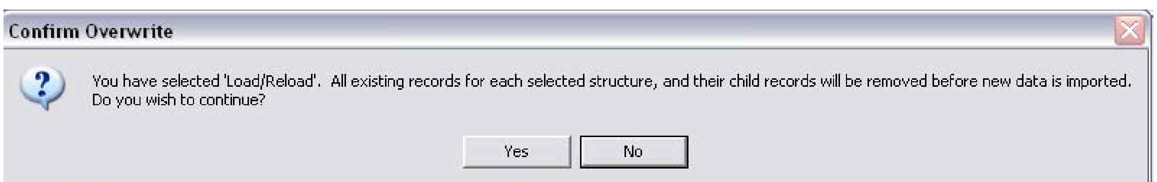



Figure 55. Confirm overwrite dialog.

## Importing from Microsoft® Excel templates

There are defined methods allowing users to import shoreline structure and storm data from pre-defined Microsoft® Excel templates. The Excel templates are placed in the main Beach-*fx* install directory under ...*\ImportTemplates\Excel_Templates*. All of the templates have a specific format which must be maintained in order for the import to succeed. It is recommended that prior to entering any data into these templates, they be copied to a different location on the user's computer, that way there will always be an empty/unmodified version of the template available for future use. Each template also has certain fields highlighted in a yellow color. These fields are required to be filled in for the import to function properly. For example, in the Profiles worksheet the column **ProfileNumber** is highlighted yellow because no Profile can be imported into Beach-*fx* without having a unique identifier in this column.

## Shoreline structures

The shoreline structures template, ImportShorelineStructures.xls, is a Microsoft® Excel workbook containing separate worksheets for importing Profiles, Reaches, Lots, and Damage Elements. Once the Excel template has been populated with data, users can invoke the appropriate import method by selecting **File->Import-> Shoreline Structures (Excel)** from Beach-*fx*'s main menu or by right-clicking on the **Shoreline Structures** section of the *Navigation Tree*. Choosing this option will display the form shown in Figure 56. Users must browse to the Excel file containing the data they wish to import. The form also contains a checkbox for each of the four structures. The import will only process data for selected structures. There is also an option to either append to or overwrite existing data. It is extremely important to note that if SBEACH runs have already been imported into Beach-*fx*, the Profiles should never be overwritten. This will corrupt the necessary linkages between the Profiles and the Shore Response Database. Lastly, the option to output a log file is available for viewing any possible errors which may have occurred during the import process. This option is not a replacement for good quality control of the data in the spreadsheet but is helpful in determining where possible problems in the data may lie.


Figure 56. Import shoreline structures (Excel).

## Storms

The storms import template (Figure 57) ImportStorms.xls, can be found within the Beach-*fx* installation directory within the import templates directory. This template is used to import the storms used in the SBEACH simulations. The Import Storms form can be launched through the **Import** menu item. The location of the import template must be provided. The user also specifies whether the imported data should be appended to the pre-existing data or if the pre-existing data should be overwritten. It is extremely important to note that once SBEACH data has been imported into the system, the **Overwrite** option should not be used, just like with Profiles. The Storms are linked to the Storm Response Database, so overwriting the storms after the initial import will break the necessary linkages, which will require another import of the SBEACH data.

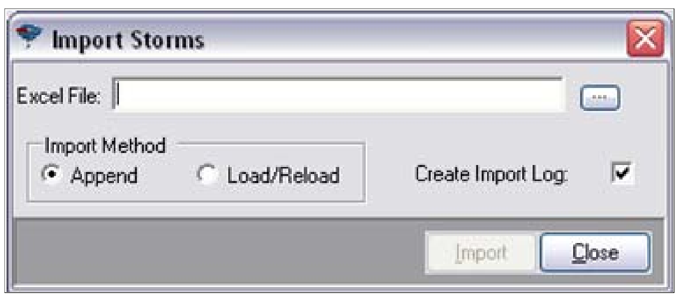



Figure 57. Import storms (Excel).

## Importing from CSV templates

The CSV files are comma delimited ASCII text files which can be opened and edited in Microsoft® Excel. Users can invoke the CSV importation methods for a Beach-*fx* object by either right-clicking on the desired object type node in the *Navigation Tree* located on the left side of the user interface or by selecting **File -> Import -> <*object_type*> (CSV**) from the application's main menu. Separate CSV import templates exist for each type of Beach-*fx* object and are provided with the installation beneath the main install directory under *...\ImportTemplates\CSV_Templates*. Before entering data into these templates, users should copy them to a separate directory in order to keep an unmodified version of each template for future use. However, users can generate these template files either by creating an empty project through the *Project Manager* and then

exporting it or by exporting an existing project and deleting the data from the generated files.

As with exporting data, certain related objects are displayed on the same import form. For example, if the user chooses to import shoreline structures, a form like the one in Figure 58 will be displayed. Because these structures are so closely related (meaning a user importing new Profiles will most likely also want to import new Reaches, etc.) the import dialog allows users to select the objects for which they want to import data by selecting the associated checkboxes. Users must then browse to and select the CSV import template file to be used for each of objects selected.

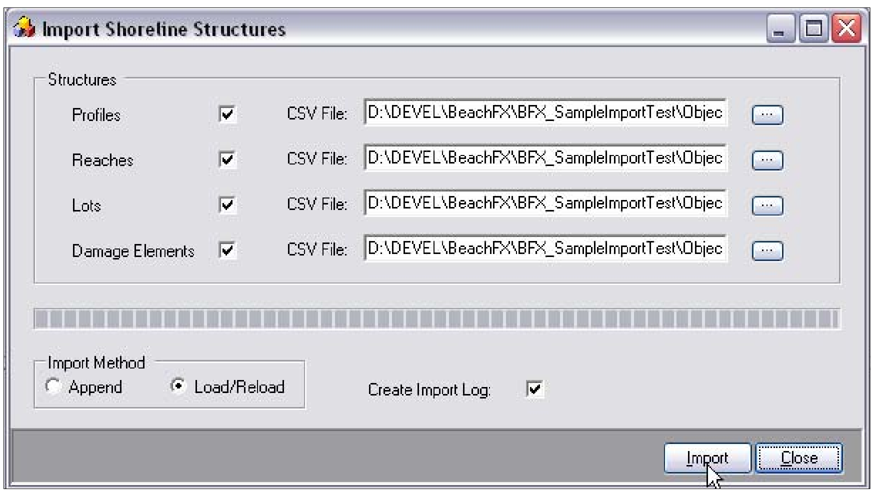



Figure 58. Importing shoreline structures (CSV).

During an import, checks are performed on each new object record to maintain data integrity as well as avoid errors that would be generated if an attempt was made to insert bad data into the database. For instance, fields used as external identifiers (i.e., ProfileNumber, ReachNumber, LotExternalID, and DamageElementExternalID) are compared to any pre-existing values to ensure uniqueness. Fields that should contain integer or decimal values are checked to make sure they do not contain strings. Required fields are checked to verify data have been entered and cells containing distribution data (P1, P2, P3—Minimum, Most Likely, Maximum) are compared to each other to make sure each value is less than or equal to the next. If an error is found that prevents a record from being imported, that record will be skipped and the next record fetched. If the user has selected the option to create an import log (selected by

default), a detailed description of the error is written to the import log. The log entry describes the type of error, the column containing the bad data, the action taken (i.e., the record was skipped or an alternate value was inserted) and the line number where the problem record is located in the import file. Once the import is complete, a message box is displayed to the user summarizing how many records were imported for each type of object and whether or not any errors occurred. If an error did occur, the import log is opened and displayed for viewing (Figure 59). The user can always choose to browse the created log file, which is located beneath the current project's root directory in a folder named *Import Log*.

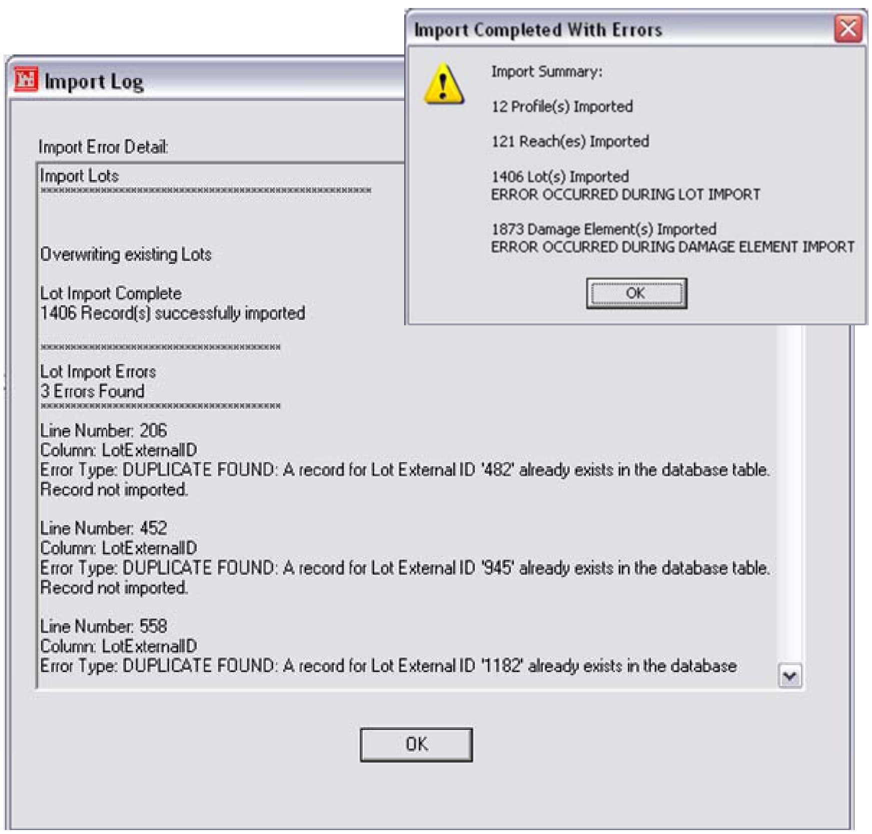



Figure 59. Import summary and detailed log.

In Figure 59, the displayed log shows that three Lots were not imported because they contained a non-unique External ID. Scrolling down in the log would show the errors that occurred during the Damage Element import. A sample of the log text is included as follows:

```
Overwriting existing Damage Elements

Damage Element Import Complete

1873 Record(s) successfully imported

****************************************

Damage Element Import Errors

535 Errors Found

****************************************

Line Number: 369

Column: DEFirstFloorElevationP1, DEFirstFloorElevationP2,
DEFirstFloorElevationValueP3

Error Type: WARNING: INVALID 3 POINT DISTRIBUTION! First
Floor Elevation Min (25) not less than or equal to First
Floor Elevation Most Likely (0) not less than or equal to
First Floor Elevation Max (0). Alternate value of 25
inserted for all 3 fields.
```

The preceeding message is a warning that the Damage Element record located at line 369 contained a First Floor Elevation P1 value of 25, and a 0 value for First Floor Elevation P2 and P3 which violated the rule that the minimum value must be less than or equal to the most likely value which must in turn be less than or equal to the maximum value. Rather than not import the record, an alternate value was inserted for the violating fields.

Note that unlike the Microsoft® Excel import, if Profiles are overwritten after the Storm Response Database has been populated, an attempt is made to reestablish the links between the Profile and the SDB records based on a comparison of the old and new Profile names. Meaning if there are existing records in the SDB tied to a Profile with the name of R1P1 and the user then imports Profile data in overwrite mode which contains a record for a Profile named R1P1, the import routine will update the corresponding records in the SDB with the newly inserted Profile's ID number. Similarly, if the user overwrites a project's Damage Functions after defining the project's Damage Function Matrix, an attempt is made to reassign the lookups based on a comparison of the old and new function names.

## Importing a project (CSV)

It is possible for users to import an entire project data set into Beach-*fx* by selecting **Import -> Project (CSV)** from Beach-*fx*'s main file menu or by right-clicking on the project's header node in the *Navigation Tree* and choosing the option from the associated context menu.

In general, users are free to move and rename the CSV files as they wish and for individual object imports doing so has no effect because the files must be manually selected each time. However, when importing a project, the import dialog (displayed in Figure 60) prompts the user to select an initialization file (.ini), rather than a CSV template. This INI file is a simple text file specifying the import file name and location for every project data structure. It is used as an alternative to forcing the user to manually specify each file separately. When a project is exported, a default INI file is generated and placed in the same directory as the CSV files containing the exported project data. The initial path for each file is relative to the INI file's location; in fact, the INI file generated on a project export expects all project-related CSV files to be located in the same directory as the INI file itself. This means a user can copy all the exported project data files to a new location without having to update the INI file. However, if the CSV files are renamed or if they are not all located in the same directory as the INI file, users will be forced to locate the missing/renamed file(s) and regenerate the INI file before they are allowed to import a project.


Figure 60. Importing a Beach-*fx* project.

When the import screen first appears, the **Import** button is disabled. When the user selects an INI file, the file is read to make sure all the referenced CSV templates exist in the specified locations. If all the CSV

files are located, the INI file is considered valid and the **Import** button is enabled. If the button remains disabled even after an INI file is selected, one or more of the template files could not be located. Users can always click on the **Create/Repair** button to view the contents of the INI file. Figure 61 demonstrates the form that is displayed when this button is clicked. If an INI file is selected that contains a bad or missing entry, the missing object's text box will be blank and marked with a red asterisk. In the example illustrated in Figure 61, the CSV file for Storm Seasons could not be found. Users can browse and select the desired file which will automatically fill in the text box with the file's location and enable the **Generate** button at the bottom of the form. Clicking on the button will update invalid portions of the INI file with the selected CSV template's file name and path. Once the file is written, users can then close out of this form, returning to the project import dialog where they can then initiate the import routine.

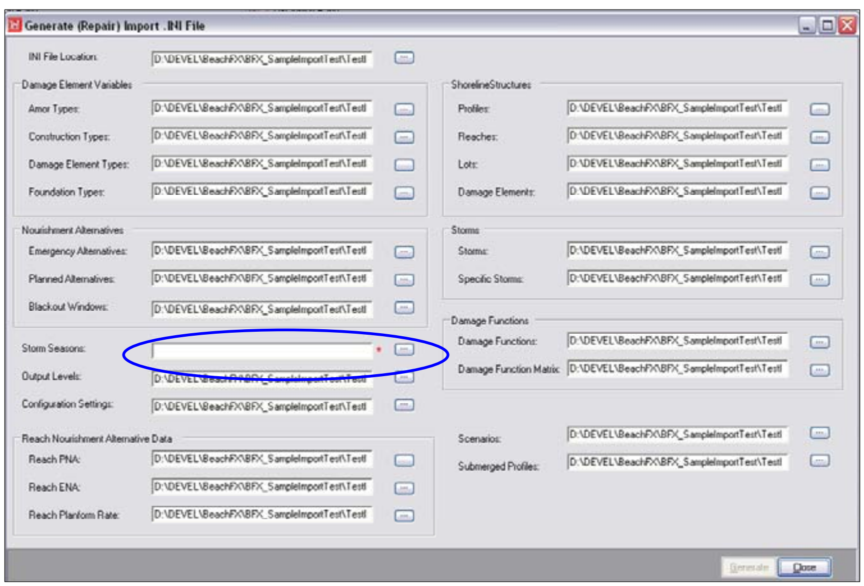



Figure 61. Generate/repair project import INI file.

The Generate (Repair) form can also be used to construct an INI file from scratch (Figure 62) by clicking on the **Create/Repair** button of the import project screen without specifying an INI file. In this situation, users would be prompted to specify a directory in which the INI file should be created. After a directory is specified, users can select a CSV template file to be used for each project component.

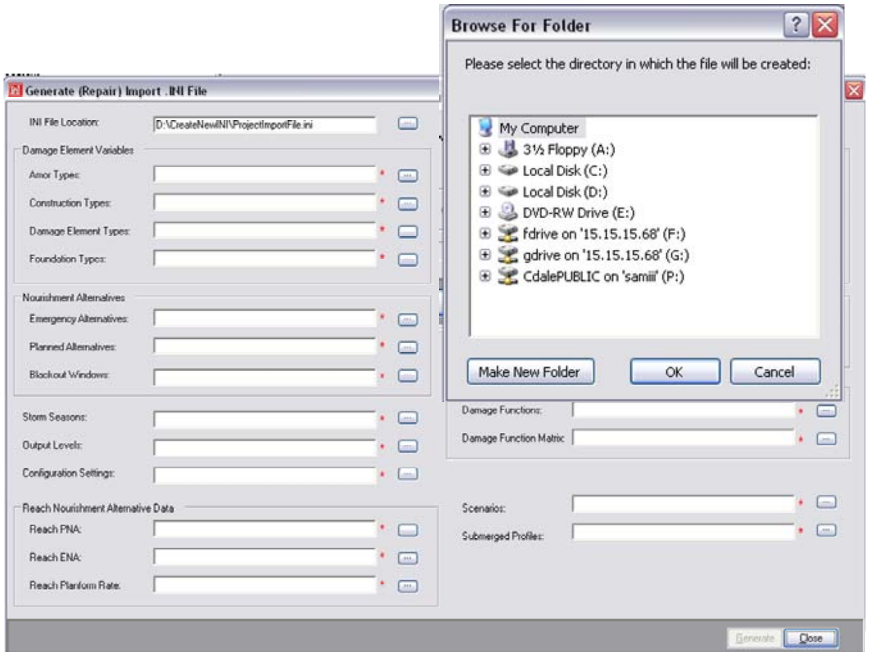


Figure 62. Create new project INI file.

## SBEACH results

The SBEACH global export output ASCII files (.dat) need to be imported into Beach-*fx* in order to populate the Storm Response Database. The SBEACH Importer (shown in Figure 63) has the ability to import .dat files in a batch process. For simplicity and easier retrieval, all .dat files should be placed in separate directory folders based on the Profile contained in the output files. For example, all results coming from SBEACH runs against Profile 1 should be in a separate folder than runs using Profile 2. All Profiles and Storms must be entered into the system prior to running the SBEACH import process. The SBEACH Import form can be launched by selecting **File -> Import -> Storm Response (SBEACH)** from the main menu. This form requires that the user specify the directory containing the SBEACH output files (.dat) from which data will be imported. Once everything has been entered pressing the **Process** button will begin importing SBEACH data. This is a lengthy process which takes approximately 1.5 hr per Gigabyte of SBEACH output files on a relatively fast computer. Progress of the import routine is displayed on the screen while the process is running.

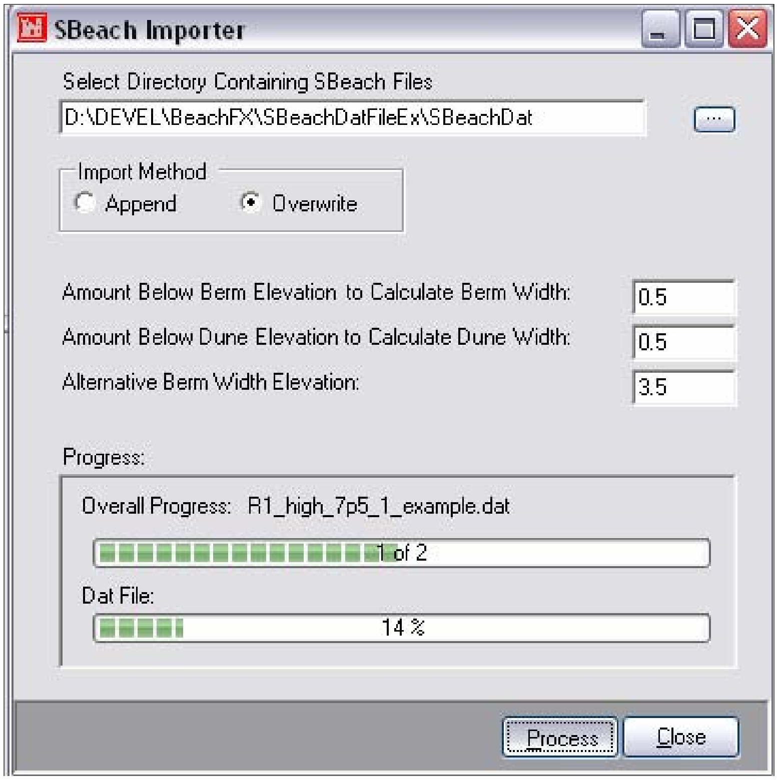



Figure 63. Import SBEACH runs.

# 10 Working with Beach-*fx* Shapefile Templates

It is possible to import data for Reaches, Lots, and Damage Elements from shapefiles as long as the corresponding attribute tables follow a specific format. Shapefile templates are provided with the Beach-*fx* install and can be found beneath the main install directory under *...\ImportTemplates\Shapefile_Templates*. The shapefiles for Reaches and Lots are defined as area or polygon shapefiles, whereas the Damage Element shapefile contains points. The attribute table fields are listed for each shapefile in Appendix A.

Users can create properly formatted shapefiles to import data into Beach-*fx* using one of three methods: template shapefiles populated with the required data, existing shapefiles modified into the necessary format, or a spreadsheet containing the required data can be joined with the shapefile to construct and populate the attribute table. Instructions for each of these methods will be discussed.

## Populating template shapefiles

As mentioned, template shapefiles are located under the Beach-*fx* install directory in *...\ImportTemplates\Shapefile_Templates*. It is recommended that the user copy the files to a separate location rather than edit the templates themselves. This way there will always be a clean copy of each of the files available for future reference.

Once the template files have been copied, open a new ArcMap project and add the appropriate template files and reference data (i.e., MrSID images). A new ArcMap project with added shapefile templates will be displayed, as shown in Figure 64.

Select the file to which data will be added first and start an editing session to begin creating the features of the shapefile. Snapping should be used for adjacent polygon features to ensure spatial continuity in the data. Users should note the direction in which vertices need to be digitized as discussed in the “Additional considerations” section.

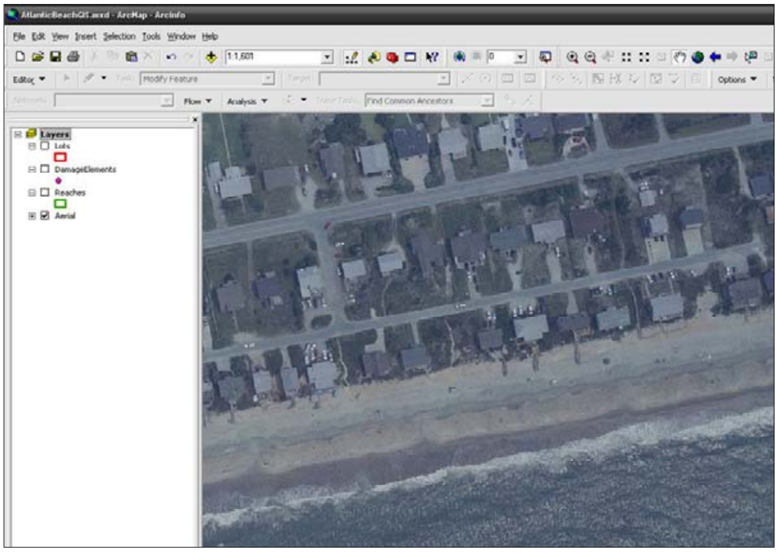


Figure 64. ARCMAP project with added files.

Use the editor window, shown in Figure 65 to populate the digitized feature's attributes. Edits should be saved frequently throughout the session to ensure work is not lost. When all features have been digitized and attributes saved, stop editing. The shapefile is now ready for import into Beach-*fx*.

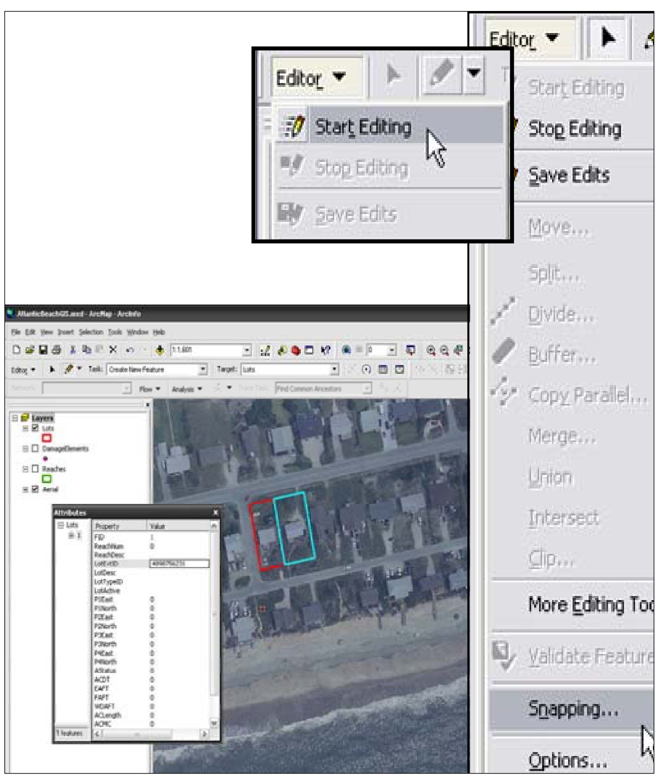



Figure 65. ARCMAP editor menu and editor window.

## Formatting existing shapefiles

Existing data shapefiles can be utilized for importing data into Beach-*fx*. The process described will use an existing parcel layer as an example. The process described can be used for any of the three types of shapefiles needed by Beach-*fx*. Open a new ArcMap project and add the object shapefile already containing spatial data and the corresponding template shapefile (in this example Lots) from the templates directory.

The fields of the Lots shapefile need to be added to the existing parcels shapefile. To accomplish this task, the merge tool will be used. Open **ArcToolbox** and navigate to **Data Management Tools > General** to select the **Merge** tool.

In the **Merge** tool window, select the parcels and lots shapefiles from the **Input Datasets** drop-down window as shown in Figure 66. A new dataset will be created. Under the output dataset, set the correct path for the storage of the new file and name the file the corresponding template name, Lots in this case, and hit **OK**.

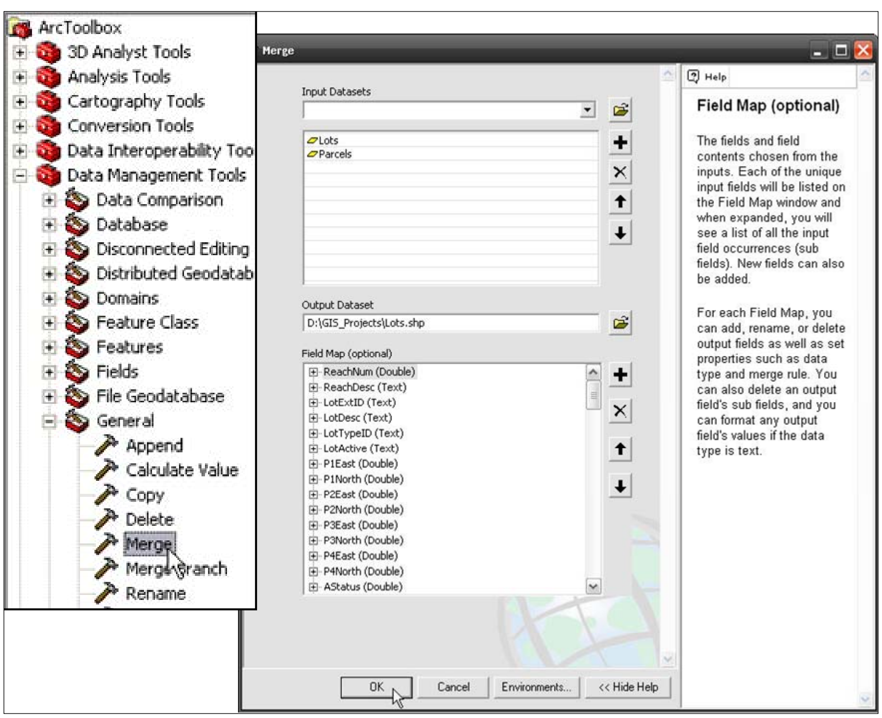



Figure 66. ARCMAP merge tools.

The new shapefile that has been automatically added to the table of contents now contains fields from the original parcels shapefile and the lots shapefile. Remove the lots template shapefile from the table of contents. Open the attribute table of the new shapefile to ensure the new fields have been added. The next step is to copy the field data from the original parcels dataset into the lots template fields. Refer to Appendix A for guidance on the data required for each field. To populate these new fields, the field calculator tool is used as shown in Figure 67. To activate the field calculator, right click on a field and select **Field Calculator**.

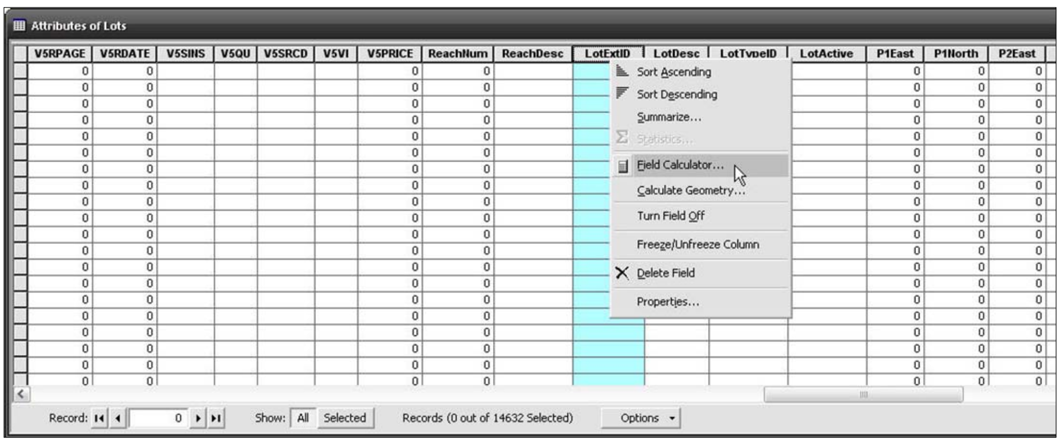



Figure 67. Created shapefile's attribute table.

The field calculator was opened for the LotExtID field. Note that the field Pin15 in the original dataset corresponds to the LotExtID field. Double click **Pin15** from the list and then click **OK** as shown in Figure 68. The information from the Pin15 field has now been copied into the LotExtID field. Repeat this process for the remaining fields. After all the data have been copied, the fields from the original shapefile need to be deleted. Right click on each field heading and select **Delete Field** as shown in Figure 69. The shapefile is now ready to be imported into Beach-*fx*.

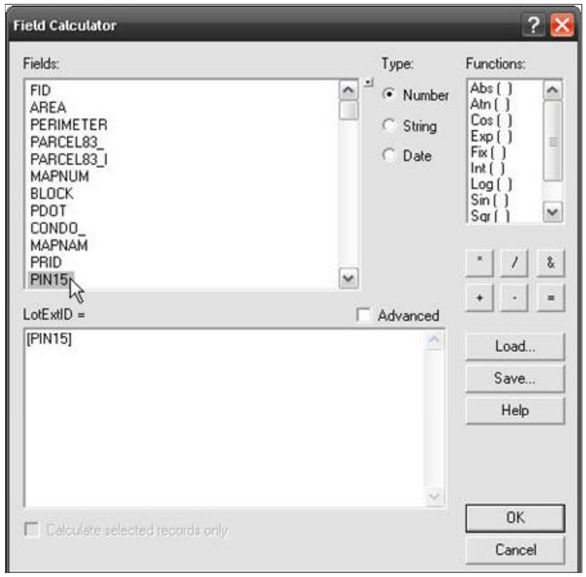


Figure 68. Field calculator.

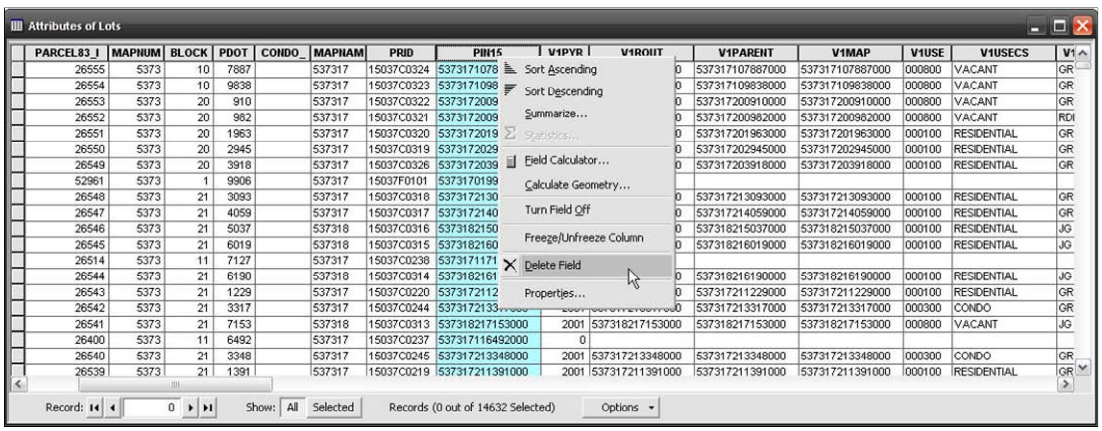


Figure 69. Deleting attribute fields.

## Joining a shapefile with a spreadsheet

Spreadsheets can be used to input the required data for the Lots, Reaches, and Damage Elements. To accomplish this, spreadsheet data will be joined with the GIS data using a common identification number. Even though the example will focus on the creation of the Damage Elements shapefile, the same process applies to each of the three shapefile types.

To begin, a new point shapefile will need to be created in ArcCatalog, this is accomplished by right clicking and select **New -> Shapefile** in the desired storage location as shown in Figure 70.

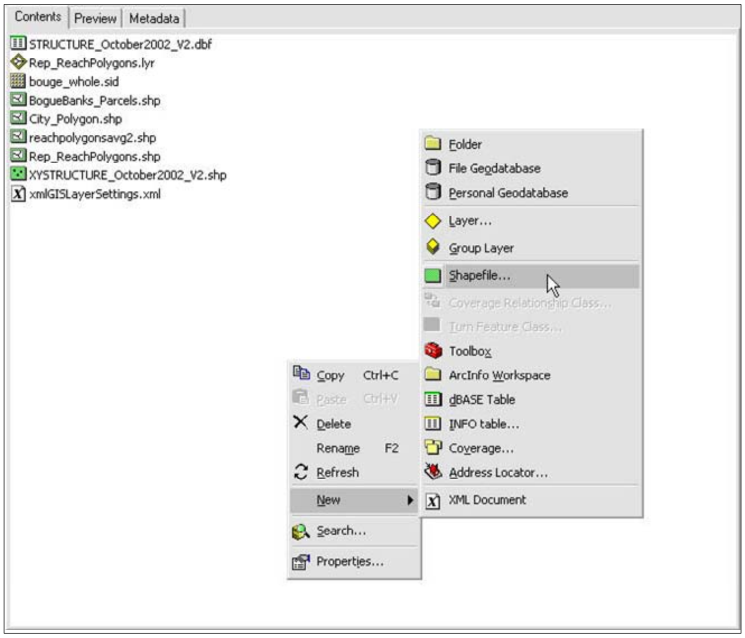



Figure 70. Create new shapefile.

Figure 71 shows the Create New Shapefile form that appears. Enter a title for the new shapefile, in this example Damage_Elements. Select **Edit** and then **Select**. Navigate to *Coordinate Systems > Projected Coordinate Systems > Geographic Coordinate Systems > North America* and then select **North America Datum 1983.prj** as this is the preferred projection. However, if the reference data are projected differently the data can be stored using the desired reference projection.

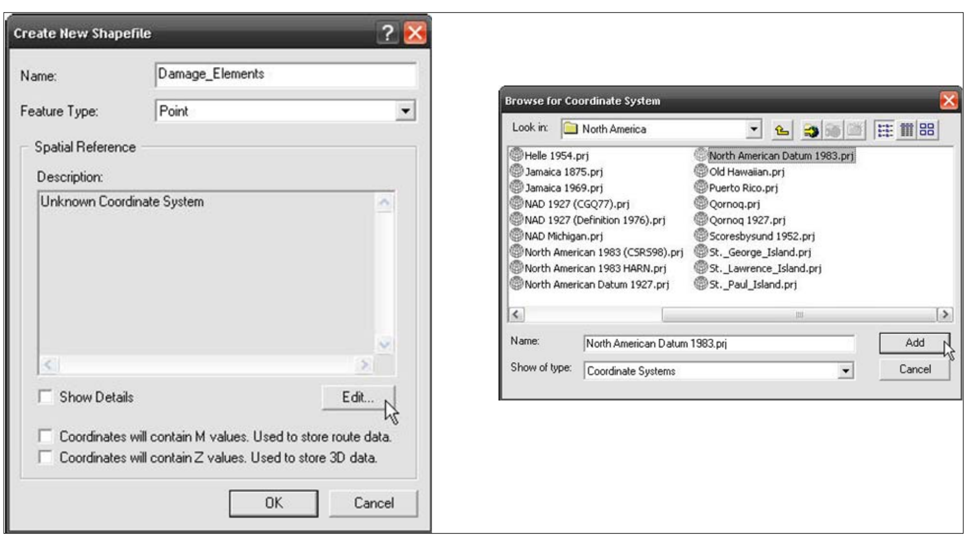



Figure 71. Assigning a projection.

Start a new ArcMap project, add the reference data, and add the new shapefile that was just created. A new column will need to be added to the attribute table. Open the attribute table and select **Options -> Add Field** to add a new column as shown in Figure 72.

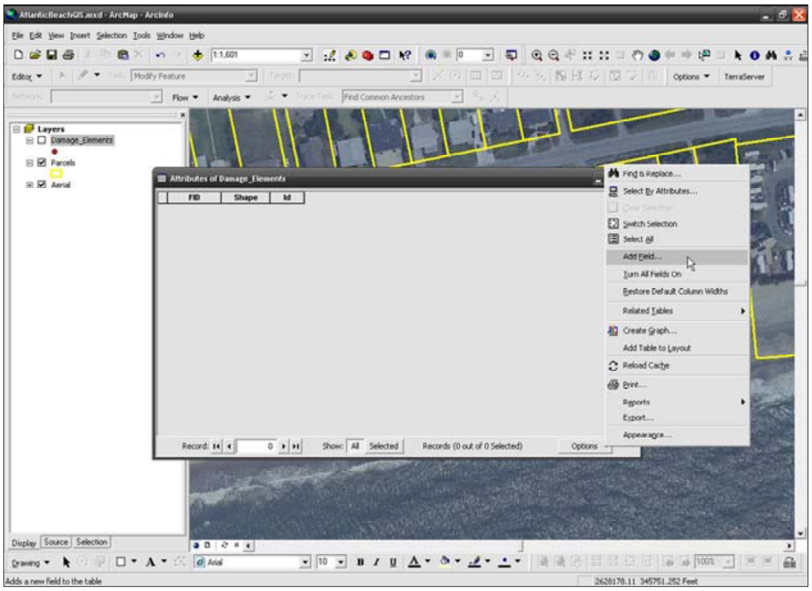


Figure 72. Add new field.

In Figure 73 a new field named UniqueID is added to the attribute table. Users can choose to name this field as they wish. However, they should be aware that this field will be used to join the spreadsheet data to the shapefile.

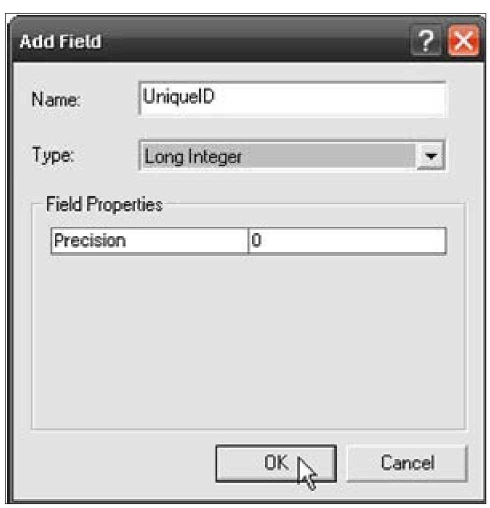



Figure 73. Add unique join field.

Begin an editing session, the Damage Element points will need to be digitized. This means that each record in the attribute table should receive a unique ID number in the UniqueID field.

Now that the shapefile with the unique ID has been created, the spreadsheet that is to be joined to can be created. The column names in the spreadsheet need to be created in the same name, type and order as in Appendix A. Add one extra field titled UniqueID or the same name as used in the point shapefile, to complete the data join. An example spreadsheet is displayed in Figure 74. Populate the spreadsheet with the required data and unique identification numbers. When complete, select **File -> Save As** and specify a CSV (Comma Delimited) file type. To join this CSV file to the created shapefile, return to the ArcMap project, click on the **Add Data** button and navigate to the location of the CSV file. Select the desired file and then **OK**. The file has now been added to the ArcMap project. To create the join, right click on the shapefile and select **Joins and Relates -> Join**, as shown in Figure 75.

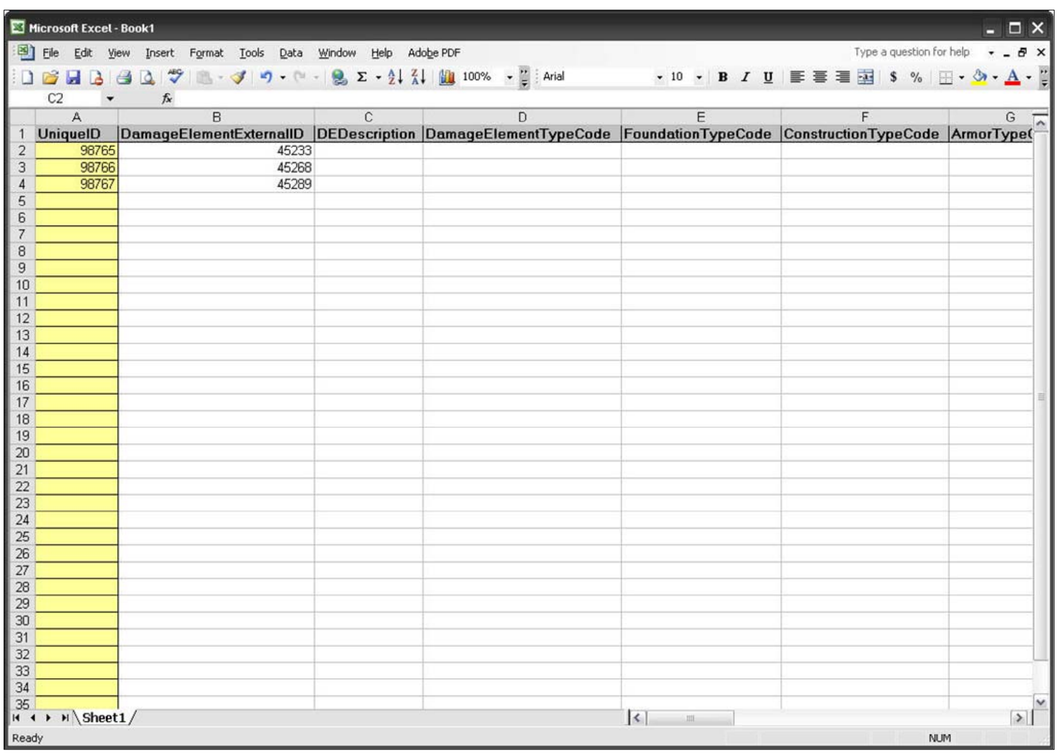



Figure 74. Example spreadsheet.

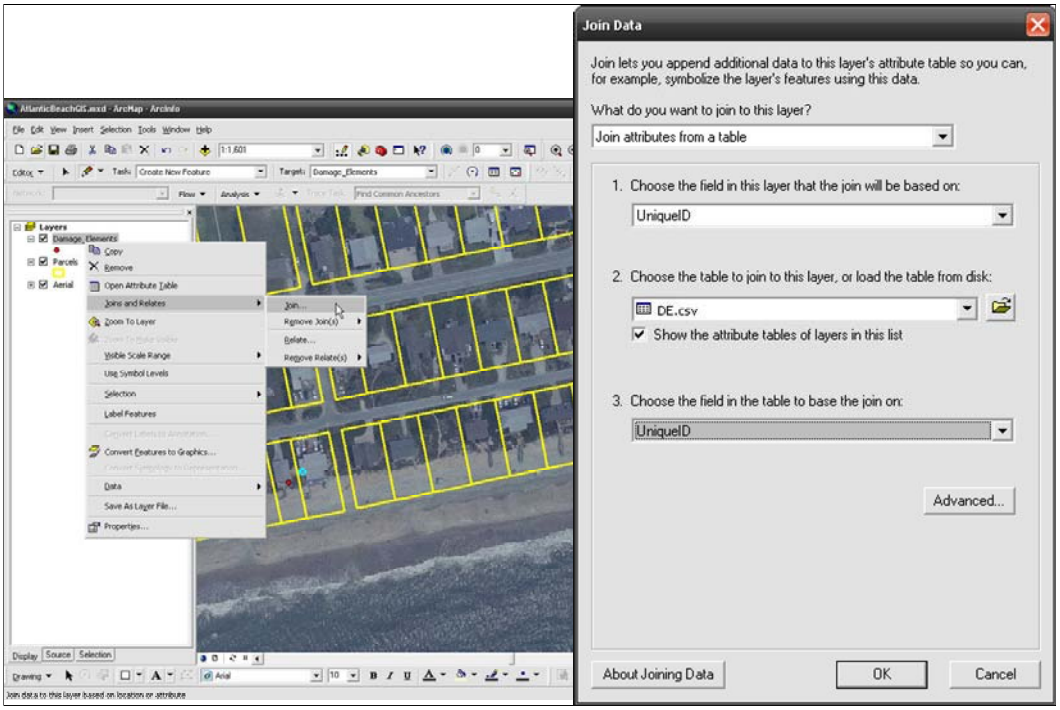


Figure 75. Joining external data to shapefile.

In the Join Data window, the first drop-down is used to identify the UniqueID field in the shapefile, the second drop-down is used to select the CSV table and the third drop-down is used to identify the UniqueID field in the CSV table.

Open the attribute table of the Damage Elements shapefile. The fields from the spreadsheet should have been added to the table. Figure 76 displays the attribute table of the Damage_Elements shapefile after completing the join.

The new shapefile will need to be exported to preserve the temporary join that was just created with the CSV spreadsheet. To export the shapefile, right click **File** in the layers list and select **Data → Export Data** from the context menu as shown in Figure 77. Select the desired export location and export the file. The final step is to bring the exported shapefile back into ArcMap and delete the UniqueID field. Once this field has been deleted, the file can be imported into Beach-*fx*.

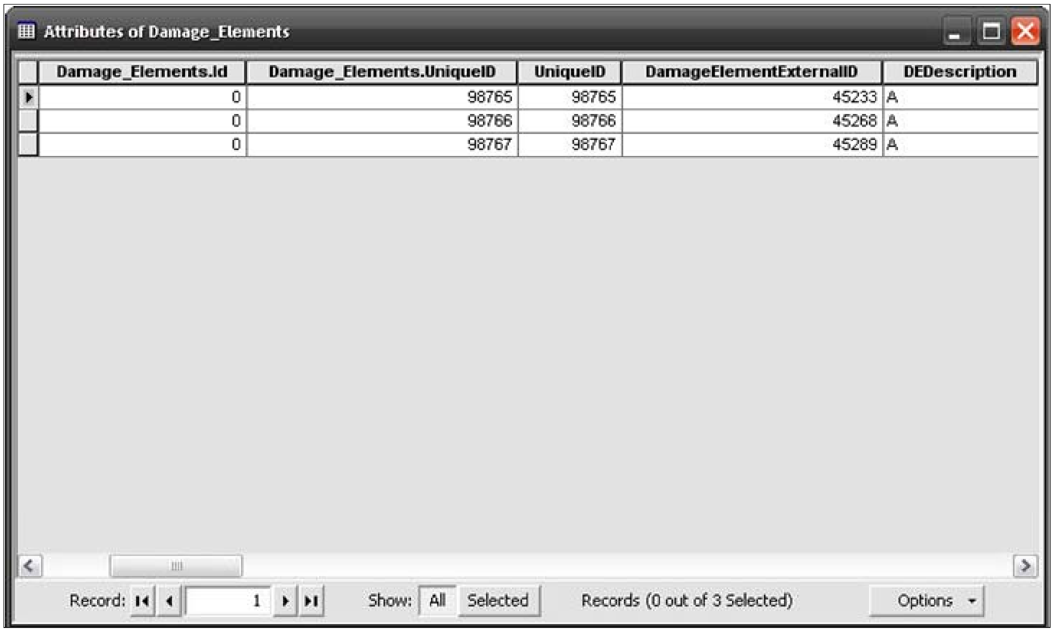



Figure 76. New attribute fields.

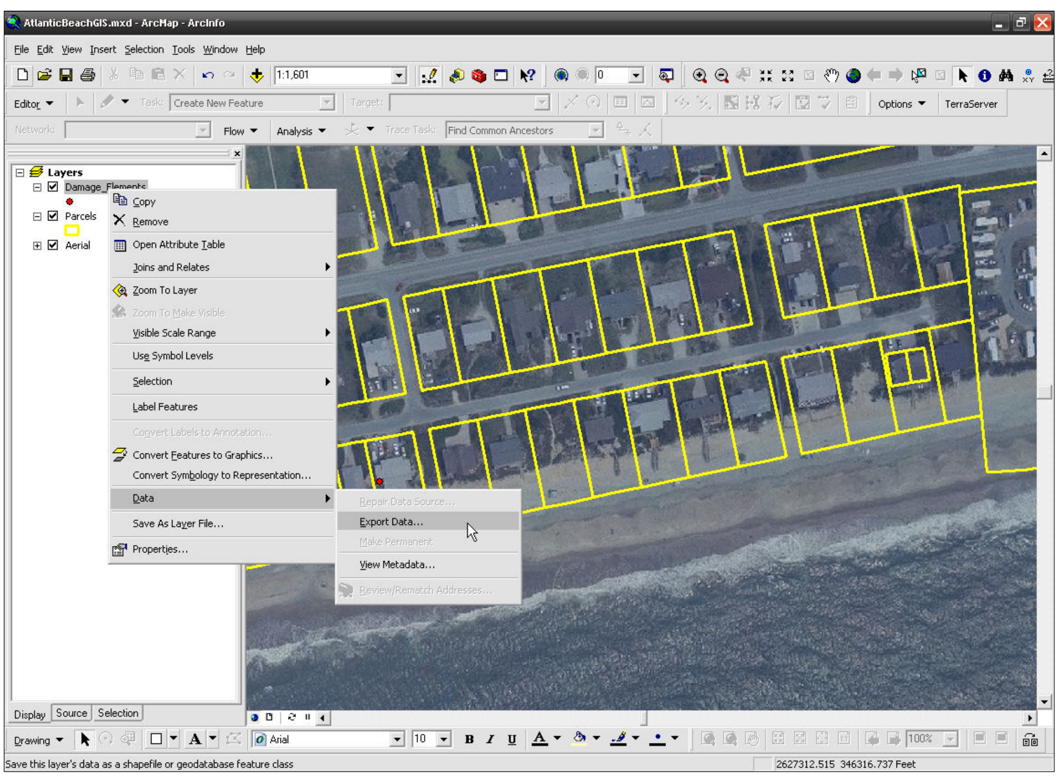



Figure 77. Export shapefile to finalize the join.

## Additional considerations

It is extremely important that the attribute tables of the shapefiles used to import structures exactly match the table formats of the template files in order for data to be imported successfully. This includes field order, field name and field data type.

Users will notice that the shape vertices (for Reaches and Lots) and point location (for Damage Elements) are included as separate fields in the attribute tables. Strictly speaking, these fields are not necessary as the points can be obtained from the .shp file, which contains all spatial data of the defined shapes. It was decided, however, to include these fields as an easy means for manipulating coordinate data and correcting possible mistakes. These fields are entirely optional, meaning data do not have to be entered by the user. The import routines first retrieve coordinate values of the vertices from the shapes themselves. If any/all coordinate data have been entered into the corresponding attribute table field(s) for an object, the manually entered data are imported for that vertex's Northing or Easting value instead.

Users should also be aware that polygons are defined by most GIS applications as being a set of clockwise points. This conflicts with Beach-*fx*'s definition of Reaches as two parallel line segments. In order for Beach-*fx* to determine which point is which (i.e., landward/seaward/start/stop), users must draw Reach vertices in the following order:

1. Landward start.
2. Landward end.
3. Shoreward end.
4. Shoreward start.

If the vertices are not drawn in this order, the import will still complete without error. However, the imported spatial data will be inserted into the wrong fields for that object record, unless the Northing/Easting values have been manually entered into the appropriate attribute table fields.

Note that data cannot be imported from shapefiles if those shapefiles are currently being used by the Beach-*fx* map display.

# References

Gravens, M. B, R. M. Males, D. A. Moser. 2007. Beach-*fx*: Monte Carlo life-cycle simulation model for estimating shore protection project evolution and cost benefit analyses. *Shore & Beach* 75(1):12-19.

Males, R. M., M. B. Gravens, D. A. Moser, and C. M. Rogers. 2007. Beach-*fx*: Life-cycle risk analysis of shore protection projects. *Proceedings 30th International Conference on Coastal Engineering*, J. M. Smith (ed.). Singapore, Japan: World Scientific Publishing Company, Inc.

Moser, D. A., M. B. Gravens, R. M. Males, M. J. Wutkowski, and C. M. Rogers. 2007. Life-cycle risk analysis approach to coastal storm damage reduction planning. *Proceedings 30th International Conference on Coastal Engineering*, J. M. Smith (ed.). Singapore, Japan: World Scientific Publishing Company, Inc.

# Appendix A: Shapefile Import Template Attribute Fields

## Reaches

| Attribute Name | Data Type | Attribute Column Description |
|---|---|---|
| ProfileNum | Integer | Profile Number |
| ProfileDesc | String | Profile Description |
| ReachNum | Integer | Reach Number |
| ReachDesc | String | Reach Description |
| ReachLen | Double | Length |
| ErosionRt | Double | Applied Erosion Rate |
| LandStartE | Double | Reach Start Point Easting |
| LandStartN | Double | Reach Start Point Northing |
| LandEndE | Double | Reach End Point Easting |
| LandEndN | Double | Reach End Point Northing |
| ShoreSE | Double | Reach Shoreward Start Point Easting |
| ShoreSN | Double | Reach Shoreward Start Point Northing |
| ShoreEE | Double | Reach Shoreward End Point Easting |
| ShoreEN | Double | Reach Shoreward End Point Northing |
| SBeachLE | Double | Reach SBeach Landward Boundary Easting |
| SBeachLN | Double | Reach SBeach Landward Boundary Northing |
| SBeachSE | Double | Reach SBeach Seaward Boundary Easting |
| SBeachSN | Double | Reach SBeach Seaward Boundary Northing |
| BermWR | Double | Berm Width Recovery Factor |
| BackBayF | String (‘TRUE’ / ‘FALSE’) | Back-Bay Flooding Flag |
| UplandW | Double | Upland Width |
| PNourishF | String (‘TRUE’ / ‘FALSE’) | Planned Nourishment Flag |
| ENourishF | String (‘TRUE’ / ‘FALSE’) | Emergency Nourishment Flag |
| FloodingT | Double | Flooding Threshold |
| EconRNum | Integer | Economic Reach Number |
| CtrlLineOf | Double | Control Line Offset |

## Lots

| Attribute Name | Data Type | Attribute Column Description |
|---|---|---|
| ReachNum | Integer | Reach Number |
| ReachDesc | String | Reach Description |
| LotExtID | String | Lot External ID |
| LotDesc | String | Lot Description |
| LotTypeID | String ('V' / 'R') | Lot Area (Residential/Vacant) |
| LotActive | String ('TRUE' / 'FALSE') | Lot Active |
| P1East | Double | Point 1 Easting |
| P1North | Double | Point 1 North |
| P2East | Double | Point 2 Easting |
| P2North | Double | Point 2 North |
| P3East | Double | Point 3 Easting |
| P3North | Double | Point 3 North |
| P4East | Double | Point 4 Easting |
| P4North | Double | Point 4 North |
| AStatus | Integer | Armoring Status |
| ACDT | Double | Distance Trigger For Armor Construction |
| EAFT | Double | Erosion Armor Failure Threshold |
| FAFT | Double | Flooding Armor Failure Threshold |
| WDAFT | Double | Wave Damage Armor Failure Threshold |
| ACLength | Double | Armor Constructed Length |
| ACMC | Double | Armor Construction Mobilization Cost |
| ACCPF | Double | Armor Construction Cost Per Foot |
| ACMT | Double | Armor Construction Mobilization Time |
| ACTPF | Double | Armor Construction Time Per Foot |

## Damage Elements

| Attribute Name | Data Type | Attribute Column Description |
|---|---|---|
| LotExtID | String | Lot External ID |
| LotDesc | String | Lot Description |
| DEExtID | String | Damage Element External ID |
| DEDesc | String | Damage Element Description |
| DEType | String | Damage Element Type Code |
| FType | String | Foundation Type Code |
| CType | String | Construction Type Code |
| AType | String | Armor Type Code |
| StructMin | Double | StructureValueP1 |
| StructML | Double | StructureValueP2 |
| StructMax | Double | StructureValueP3 |
| ContentMin | Double | ContentsValueP1 |
| ContentML | Double | ContentsValueP2 |
| ContentMax | Double | ContentsValueP3 |
| DEWidth | Double | DEWidth |
| DELength | Double | DELength |
| FFEMin | Double | DEFirstFloorElevationP1 |
| FFEML | Double | DEFirstFloorElevationP2 |
| FFEMax | Double | DEFirstFloorElevationP3 |
| NumOfFloor | Integer | DENumberOfFloors |
| TBuildMin | Double | TimeToRebuildP1 |
| TBuildML | Double | TimeToRebuildP2 |
| TBuildMax | Double | TimeToRebuildP3 |
| BuildNum | Integer | NumberOfTimesRebuildingAllowed |
| DEActive | String (‘TRUE’/’FALSE’) | Damage Element Active Flag |
| RepPtN | Double | Representative Point Northing |
| RepPtE | Double | Representative Point Easting |

# Appendix B: General Description of Output Files

## File names, types, and locations

Files are located in the same directory as the selected input database (IDB file) that is associated with the project being run. Output files are prefixed by the name of the Scenario that has been run, and have a suffix that describes the data in the file. The only exception to this is an error file generated each time the Beach-*fx* simulation is run, that has the name format BeachFxnnnnnnnnnnnn.err, where nnnnnnnnnnnn is a number generated based on the date and time that the simulation is run, e.g., BeachFx200719131401.err. Note that this file is always generated. Note also that the error file reports errors in processing that may be encountered before other output files are generated. This file should be examined whenever there is an unexpected termination of the program, as it may provide insight into the problem.

Output files are of two types, ASCII or CSV (Excel compatible). ASCII-format files generated by Beach-*fx* have the extensions .txt, .prn, and .err. If the CSV output file has fewer than 65,000 (roughly) rows, it can generally be imported into an Excel worksheet directly, simply by clicking on it. If the file is larger, it needs to be imported into a database program such as Access to examine the outputs in detail.

## User control of output file generation

The presence or absence of most (but not all) files is subject to user control. Beach-*fx* currently generates 43 output files, of which 38 are under user control. A user can specify whether or not the file is to be created, and, in some cases, the frequency at which data are added to the file. However, there is no user specification of level of detail provided within a given output data file, with the exception of the .dbg (debug) file. The debug file is always present, but will be small unless the user turns on output, in which case it can grow large. File sizes are either constant (roughly independent of any user settings), or, more commonly, linear with the number of iterations. Certain files are output at each time-step or

at the end of each year, thus the file size will vary with the duration of the simulation and the user setting for the time-step.

The following files are not under user control, and will be created for every run:

| FileSuffix | Description | Generation |
|---|---|---|
| .prn | Run summary results | Once per run |
| .err | Error messages | Once per run |
| .dbg | Debug messages | Once per run |
| _Iteration.csv | Iteration totals (e.g. number of storms, Emergency Nourishment, costs, etc.) | Once per iteration |
| _Message.csv | Messages that should be examined by the user | When warnings or critical messages are detected |
| _Memory.csv | Memory usage by the program | At the end of each iteration |

The following files are under user control, as specified by the output control settings that can be changed through the user interface (*Navigation Tree* item Output Options). The Setting column indicates a typical setting for output control for each file. Development indicates that the file is typically examined during initial database development and testing (1 or 2 iteration runs), but the file is usually suppressed during production runs. Debugging indicates that the file is usually used only for testing, with the assistance of the development team. Production indicates that the files can be generated for production runs, if file sizes do not grow too large. Size columns indicates those files that may grow quite large (L) with a large number of iterations. Note that the simulation runs faster when fewer output files are generated, in particular the large files. Thus, once the user is confident of behavior, it is advisable to run with a minimal set of output files. It is always possible to rerun a Scenario with additional output files, if needed.

| FileSuffix | Description | Generation | Setting | Size |
|---|---|---|---|---|
| _AnimationSetup.csv | Setup file for 3D planform animation | One time, after reading input data | Off | |
| _CoordinateChecks.txt | Data Checking for Reach / Lot / Damage Element Coordinates | One time, after reading input data | Development | |
| _Damage.csv | Structure and Contents damage each damage occurrence | Within simulation when damages exist | Early production testing | L |
| _Damage2.csv | Structure and Contents damage each damage occurrence alternate format | Within simulation when damages exist | Early production testing | L |
| _DamageElementEcho.csv | Echo of input Damage Element information | One time, after reading input data | Development | |
| _DamageElementErrors.csv | Error messages from coordinate checking for Damage Elements | One time, after reading input data | Development | |
| _DamageValueHistory.csv | Structure and Contents value by Damage Element over time | Within simulation on change in Damage Element value due to storm or rebuilding | Early production testing | L |
| .dbg | Debug file Information – not generally user-friendly | Throughout simulation | Debugging | L |
| _DECondemnation.csv | Damage Element condemnation information | On event if condemnation takes place | Production | |
| _DESummary1.csv and _DESummary2.csv | Value summaries by Damage Element Type (DESummary1) and type and Reach (DESummary2) | One time, after reading input data | Production | |

| FileSuffix | Description | Generation | Setting | Size |
|---|---|---|---|---|
| .ech | Echo of input data | One time, after reading input data | Development | |
| _EmergencyNourishment.csv | Detailed information on each Reach Emergency Nourishment | Within simulation | Production | |
| _ENSummary.csv | Emergency Nourishment summary (volume, cost) by Reach | End of simulation statistics | Production | |
| .evn | Event output | Within simulation on relevant events | Debugging | |
| _LandLoss.csv | End of year Reach locations, dimensions, and statistics | End of year | Production | |
| _LookupQuality.csv | Report of shore response lookups that fall outside tolerances | Exception report on storm response event | Development | |
| _LotCondemnation.csv | Report of Lots marked as unbuildable | On event causing Lot to be unbuildable | Production | |
| _LotEcho.csv | Echo of Lot input information | One time, after reading input data | Development | |
| _LotErrors.csv | Error report for Lot location, area outside of tolerance limits | One time, after reading input data | Development | |
| _MorphologyTimeLine and _MorphologyConstants.csv | Post-processing 2D animation driver files | Within simulation on relevant events | Development | |
| Nourishment.csv | Report of costs, volume for planned and Emergency Nourishment event | Within simulation on planned or Emergency Nourishment events | Production | |
| _NourishmentCalendar.csv | Start and finish of | Within | Production | |

| FileSuffix | Description | Generation | Setting | Size |
|---|---|---|---|---|
| | Planned and emergency plus blackout windows | simulation | | |
| _PlannedNourishment.csv | Information on each Planned Nourishment cycle | Within simulation on checking and ending Reach nourishment | Production | |
| ProfileResponse.dbg | Text output for reviewing details of storm-induced morphology change calculations | Within simulation on storm-induced morphology change event | Debugging | L |
| _Reach.csv | Information by Reach at constant times within simulation (can get very large) | Within simulation at user-selectable report interval | Debugging | L |
| _Reach2.csv | End of year report on Reach profile | Within simulation, end of year | Production | |
| _ReachChangeStatistics.csv | Statistical summary of Reach profile changes | End of simulation | Production | |
| _ReachIteration.csv | Totals by Reach for damages, Emergency Nourishment, profile location at end of iteration | End of each iteration | Production | |
| _PlannedNourishmentReach.csv | Information on each Reach Planned Nourishment | Within simulation | Production | |
| _ReachStatistics.csv | End of simulation statistics on each Reach (erosion, damages, Emergency Nourishment, etc.) | End of simulation | Production | |
| _ReachYearlyDamages.csv | Structure and contents damages by Reach/year | Within simulation on year | Production | |
| _ReachYearlyDamagesByType.csv | Structure and contents damages by Reach/year, | Within simulation on year | Production | |

| FileSuffix | Description | Generation | Setting | Size |
|---|---|---|---|---|
| | Damage Element Type | when damages exist | | |
| _Rebuilding.csv | Report by Damage Element of initial and rebuilt structure and contents value | Within simulation on completion of rebuilding | Production | |
| _StormGeneration.csv | Information on each generated storm | Within Simulation on storm event | Production | |
| _StormIteration.csv | Statistics on number of storms by iteration, season, storm type | End of iteration | Production | |
| _StormResponse.csv | Pre and post-storm morphology information | Within Simulation on storm event | Early production testing | |

## Individual output file descriptions

Descriptions of the various files are given here. Some files that are oriented primarily to development and future features are described only briefly, and it should be noted that there may be inconsistencies between the examples of different output file types, which may be drawn from different runs. Information presented is illustrative in nature only.

### *Debug – dbg*

Information is variable, used primarily for debugging. This file should generally be set to **Off**.

### *Echo – ech*

ASCII echo of input information read from the database, used in debugging.

### *Event – evn*

Chronological list of events processed by simulation, giving iteration, time within iteration, internal event identifier if available, event type, and additional information on some events. This is useful for tracking the order of processing. Can be suppressed entirely by user, and is generally used for debugging.

```
1      0.000     0 YS  Year: 2000
1    366.000     0 YS  Year: 2001
1    731.000     0 YS  Year: 2002
1   1096.000     0 YS  Year: 2003
1   1461.000     0 YS  Year: 2004
1   1827.000     0 YS  Year: 2005
1   2039.000     0 CSN  Needed: 991475.2292 Threshhold: 500000.0000
1   2039.010     0 SSN  Cycle:  1 MobCost: 400000.000
1   2039.020     0 CRN R-1   1
1   2039.020     0 SRN R-1   1
1   2047.955     0 ERN R-1   1 start:   2039.020 duration:      8.935 v: 268057.235
1   2047.965     0 CRN R-2   2
1   2047.965     0 SRN R-2   2
1   2057.129     0 ERN R-2   2 start:   2047.965 duration:      9.163 v: 274898.663
1   2057.139     0 CRN R-3   3
1   2057.139     0 SRN R-3   3
1   2064.571     0 ERN R-3   3 start:   2057.139 duration:      7.432 v: 222964.778
1   2064.581     0 CRN R-4   4
1   2064.581     0 SRN R-4   4
1   2072.097     0 ERN R-4   4 start:   2064.581 duration:      7.517 v: 225497.380
1   2104.949   314 ST  WCe_19641004_H2      2.760 S:  0
1   2125.949   314 REC
1   2192.000     0 YS  Year: 2006
1   2557.000     0 YS  Year: 2007
1   2579.996     2 REB  R-2      20 282 Beachside Garage 314
1   2580.412    24 REB  R-4     139 Lakefront Pool House 314
```

Event codes of interest are:

| ST | Storm | Reports StormID, storm name, Peak Surge Plus Tide, whether storm is selected by user (1) or generated (0) |
|---|---|---|
| CSN | Check Scheduled Nourishment Cycle | With-Project |
| SSN | Start Scheduled Nourishment Cycle | With-Project |
| CRN | Check Reach Nourishment | With-Project |
| SRN | Start Reach Nourishment | With-Project |
| ERN | End Reach Nourishment | With-Project |
| SER | Start Emergency Renourishment | Reports Reach Number |
| EER | End Emergency Renourishment | Reports Reach Number |
| YS | Year Start | Reports Year |
| REC | Recovery | Reports storm ID of corresponding storm |
| REB | Rebuilding | Reports Lot index, Reach index, Damage Element index (not user-friendly) |
| PRB | Partial rebuild | Partial structure rebuild (at start of storm during rebuilding) |
| SBW | Start nourishment blackout window | |
| EBW | End nourishment blackout window | |

***Output - .prn***

ASCII file providing summary statistical information at end of simulation, giving number of observations, average, standard deviation, maximum, minimum, iteration at which maximum occurs, and iteration at which minimum occurs, for a variety of different measures:

- Average Annual Erosion by Reach Statistics. Statistics derived from calculating, for each Reach and iteration, the recession of the shoreline position between the start and end of the iteration, and dividing by the number of years in the iteration, giving a total Reach erosion per iteration, which is then accumulated statistically.

- Total Structure and Contents Damage Statistics. Statistics obtained by calculating present value of total damages in iteration (all Damage Elements, all years of iteration), summing statistically over all iterations.

- Reach Structure and Contents Damage Statistics. Same as above, excepted calculated on a Reach-by-Reach basis.

- Emergency Nourishment Mobilization and Placement Costs.
- Emergency Nourishment and Reach Geometry Information By Reach.
- Dune Height. Averaged over all time-steps, all iterations.
- Dune Width. Averaged over all time-steps, all iterations.
- Berm Width. Averaged over all time-steps, all iterations.
- Number of Emergency Nourishments. Average number per iteration.
- Emergency Nourishment Cost, Present valued total cost in iteration.
- Emergency Nourishment Volume. Total volume of nourishment in iteration.
- Storm Generation Statistics by Storm Type and Season. Over all iterations.
- Storm Interarrival Violations by Season. Over all iterations.

***AnimationSetup.csv***

This file is used in conjunction with three-dimensional (3D) visualization/animation software, not yet fully integrated into the Beach-*fx* environment, thus is primarily for developmental purposes and should be set to **Off**.

***CoordinateChecks.txt***

This file is used to report locational checks made for Lot and Damage Element locations, useful during data development. It provides information on Lot location tests: whether or not Lot quadrilaterals are properly formed, whether they are above or below user-defined maximum and minimum Lot sizes (a means of flagging bad coordinates), and which Reaches each corner of the Lot quadrilateral is in. For Damage Elements, tests are made as to whether the coordinates of the Damage Element fall within the assigned Reach quadrilateral, and the location of the Damage

Element along the cross-shore coordinate line is reported. A brief sample of the initial output of this file is presented next.

```
Lot Quadrilateral - Size in Acres, Properly Formed?
   1      R-1 Lot:     1   1  21.42 LOT AREA FLAG
   2      R-2 Lot:     2   2   0.23 LOT AREA FLAG
   2      R-2 Lot:     3   3   0.24 LOT AREA FLAG
   2      R-2 Lot:     4   4   0.22 LOT AREA FLAG
   2      R-2 Lot:     5   5   0.34 LOT AREA FLAG
   2      R-2 Lot:     6   6   0.70 LOT AREA FLAG
   2      R-2 Lot:     7   7   0.31 LOT AREA FLAG
   2      R-2 Lot:     8   8   0.71 LOT AREA FLAG
```

### Damage.csv and _Damage2.csv

These output files provide detailed information useful for checking the damage calculations for a Damage Element. The two files (_Damage and _Damage2) provide similar information, but in two different formats. Each file reports the damage calculations that take place for a Damage Element after a storm. _Damage.csv generates two lines for each Damage Element (1 for the structure, and the 2nd line for the contents) while _Damage2.csv reports a single line (with both structure and contents information) for each Damage Element. The user in general will prefer one or the other of these formats. Since these files can grow quite large, it is usually advisable to set these files to **On**, only during testing or verification of behavior, and **Off** in normal production usage. An example of portions of the _Damage.csv file is shown here.

The following two output file segments gives basic information on the Damage Element.

| Iteration | Time | StormID | LotID | DE_ID | DEType | Construction | Foundation | Armor | Reach# | Reach | Economic Reach |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3900.959 | 385 | 1 | 2 | GAZEBO | WOOD | Pile | Null | 1 | R-1 | 1 |
| 1 | 3900.959 | 385 | 1 | 2 | GAZEBO | WOOD | Pile | Null | 1 | R-1 | 1 |
| 1 | 3900.959 | 385 | 1 | 3 | MFR1 | MAS | Pile16 | Null | 1 | R-1 | 1 |
| 1 | 3900.959 | 385 | 1 | 3 | MFR1 | MAS | Pile16 | Null | 1 | R-1 | 1 |
| 1 | 3900.959 | 385 | 1 | 4 | MFR1 | MAS | Pile16 | Null | 1 | R-1 | 1 |
| 1 | 3900.959 | 385 | 1 | 4 | MFR1 | MAS | Pile16 | Null | 1 | R-1 | 1 |
| 1 | 3900.959 | 385 | 1 | 5 | MFR1 | MAS | Pile16 | Null | 1 | R-1 | 1 |
| 1 | 3900.959 | 385 | 1 | 5 | MFR1 | MAS | Pile16 | Null | 1 | R-1 | 1 |
| 1 | 3900.959 | 385 | 1 | 6 | MFR1 | MAS | Pile16 | Null | 1 | R-1 | 1 |

| DE_ID | Type | Description | External ID | BB Flood |
|---|---|---|---|---|
| 2 | S | Pinnacle Port Gazebo | 11 | 0 |
| 2 | C | Pinnacle Port Gazebo | 11 | 0 |
| 3 | S | Pinnacle Port B6 | 7 | 0 |
| 3 | C | Pinnacle Port B6 | 7 | 0 |
| 4 | S | Pinnacle Port B9 | 10 | 0 |
| 4 | C | Pinnacle Port B9 | 10 | 0 |
| 5 | S | Pinnacle Port B7 | 8 | 0 |
| 5 | C | Pinnacle Port B7 | 8 | 0 |
| 6 | S | Pinnacle Port B5 | 6 | 0 |
| Type = S = structure, C = contents. BBFlood indicates whether or not back-bay flooding is present. | | | | |

This output file segment contains information on the damage driving parameters (DP). If there is damage, then the combined code is non-zero.

| DE_ID | FF Elevation | DP Flood | DP Wave | DP Erosion | Flood Water Level | Wave Water Level | Combined Code | Combined Impact |
|---|---|---|---|---|---|---|---|---|
| 2 | 13.626 | -13.626 | -13.626 | 0 | 0 | 0 | 0 | 0 |
| 2 | 13.626 | -13.626 | -13.626 | 0 | 0 | 0 | 0 | 0 |
| 3 | 13.861 | -3.868 | -3.868 | 0 | 9.994 | 9.994 | 4 | 0.0901 |
| 3 | 13.861 | -3.868 | -3.868 | 0 | 9.994 | 9.994 | 4 | 0.3973 |
| 4 | 28.03 | -19.108 | -19.108 | 0 | 8.922 | 8.922 | 0 | 0 |
| 4 | 28.03 | -19.108 | -19.108 | 0 | 8.922 | 8.922 | 0 | 0 |
| 5 | 27.205 | -21.436 | -21.436 | 0 | 5.769 | 5.769 | 0 | 0 |
| 5 | 27.205 | -21.436 | -21.436 | 0 | 5.769 | 5.769 | 0 | 0 |
| 6 | 12.488 | -2.752 | -2.752 | 0 | 9.737 | 9.737 | 4 | 0.1171 |

This output file segment shows the individual factors that go into the calculation of the combined impact.

| DE_ID | Flood Factor | Wave Factor | Erosion Factor |
|---|---|---|---|
| 2 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 |
| 3 | 0 | 0.0901 | 0 |
| 3 | 0 | 0.3973 | 0 |
| 4 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 |
| 6 | 0 | 0.1171 | 0 |

This output file segment shows the value change (if any) associated with the damages.

| DE_ID | Pre-Storm Value | Post-Storm Value | Combined Value Loss | Flood Loss | Wave Loss | Erosion Loss | Combined Value Loss PV | Flood Loss PV | Wave Loss PV | Erosion Loss PV | PV Factor |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1009 | 1009 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.747708 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.747708 |
| 3 | 2041596 | 1857556 | 184040.1 | 0 | 184040.1 | 0 | 137608.3 | 0 | 137608.3 | 0 | 0.747708 |
| 3 | 1020798 | 615261.6 | 405536.4 | 0 | 405536.4 | 0 | 303223 | 0 | 303223 | 0 | 0.747708 |
| 4 | 328874.2 | 328874.2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.747708 |
| 4 | 164437.1 | 164437.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.747708 |
| 5 | 487157.7 | 487157.7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.747708 |
| 5 | 243578.9 | 243578.9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.747708 |
| 6 | 1975031 | 1743719 | 231311.8 | 0 | 231311.8 | 0 | 172953.7 | 0 | 172953.7 | 0 | 0.747708 |
| 6 | 987515.4 | 550716.3 | 436799 | 0 | 436799 | 0 | 326598.3 | 0 | 326598.3 | 0 | 0.747708 |

The final columns of the output file show the damage functions used for the calculations, and the associated lookups (DFWave0, DFWave1, DFWave2) from each of the functions that are used to develop the triangular distribution.

| DE_ID | FWave0 | DFWave0 | FWave1 | DFWave1 | FWave2 | DFWave2 |
|---|---|---|---|---|---|---|
| 2 | WAVEPSMIN | 0 | WAVEPSMOSTLIKELY | 0 | WAVEPSMAX | 0 |
| 2 | WAVEPCMIN | 0 | WAVEPCMOSTLIKELY | 0 | WAVEPCMAX | 0 |
| 3 | WAVEPSMIN | 0.3 | WAVEPSMOSTLIKELY | 0.4 | WAVEPSMAX | 0.5 |
| 3 | WAVEPCMIN | 0.3 | WAVEPCMOSTLIKELY | 0.4 | WAVEPCMAX | 0.5 |
| 4 | WAVEPSMIN | 0 | WAVEPSMOSTLIKELY | 0 | WAVEPSMAX | 0 |
| 4 | WAVEPCMIN | 0 | WAVEPCMOSTLIKELY | 0 | WAVEPCMAX | 0 |
| 5 | WAVEPSMIN | 0 | WAVEPSMOSTLIKELY | 0 | WAVEPSMAX | 0 |
| 5 | WAVEPCMIN | 0 | WAVEPCMOSTLIKELY | 0 | WAVEPCMAX | 0 |
| 6 | WAVEPSMIN | 0.3 | WAVEPSMOSTLIKELY | 0.4 | WAVEPSMAX | 0.5 |
| 6 | WAVEPCMIN | 0.3 | WAVEPCMOSTLIKELY | 0.4 | WAVEPCMAX | 0.5 |

***DamageElementEcho.csv***

This output file provides an echo of information for each Damage Element, together with the calculated SBEACH reference line position for the DE, and a status check to insure that necessary information is appropriately provided.

Basic identification and association information (if a logical error is found, the status column will provide additional information).

| Status | Description | Reach # | Reach | Economic Reach | Lot ID | DE_ID | External ID |
|---|---|---|---|---|---|---|---|
| | Pinnacle Port B1 | 1 | R-1 | 1 | 1 | 10 | 1 |
| | Pinnacle Port B8 | 1 | R-1 | 1 | 1 | 11 | 9 |
| | Pinnacle Port Pool | 1 | R-1 | 1 | 1 | 12 | 12 |
| | Pinnacle Port Pool2 | 1 | R-1 | 1 | 1 | 13 | 13 |
| | Pinnacle Port Tennis Court | 1 | R-1 | 1 | 1 | 14 | 14 |
| | 278 Beachside | 2 | R-2 | 1 | 2 | 15 | 15 |
| | 278 Beachside Walkway | 2 | R-2 | 1 | 2 | 16 | 16 |
| | 280 Beachside | 2 | R-2 | 1 | 3 | 17 | 17 |
| | 280 Beachside Garage | 2 | R-2 | 1 | 3 | 18 | 18 |

Damage Element Type associations and location.

| Description | DE Type | Construction | Foundation | Armor | DE Width | DE Length | Easting | Northing | SBeach |
|---|---|---|---|---|---|---|---|---|---|
| Pinnacle Port B1 | MFR1 | MAS | Pile16 | Null | 115 | 44 | 1498648 | 463653.7 | 318.2 |
| Pinnacle Port B8 | MFR1 | MAS | Pile16 | Null | 44 | 360 | 1499547 | 463318.7 | 208.4 |
| Pinnacle Port Pool | POOL | MAS | Slab | Null | 40 | 50 | 1499351 | 463355.8 | 264.5 |
| Pinnacle Port Pool2 | POOL | MAS | Slab | Null | 55 | 25 | 1499051 | 463717.6 | 78.3 |
| Pinnacle Port Tennis Court | TENNIS | MAS | Slab | Null | 200 | 116 | 1499187 | 463766.4 | -27 |
| 278 Beachside | SFR1 | MAS | Pile16 | Null | 20 | 85 | 1499996 | 463071.6 | 309.3 |
| 278 Beachside Walkway | WALK | WOOD | Pile | Null | 4 | 51 | 1499958 | 463017.7 | 374.3 |
| 280 Beachside | SFR1 | MAS | Pile16 | Null | 20 | 83 | 1500027 | 463053.9 | 311 |
| 280 Beachside Garage | SFR1 | MAS | Slab | Null | 20 | 20 | 1500056 | 463116.4 | 242.1 |

Damage Element structure and contents value triangular distribution parameters.

| Description | StructureP1 | StructureP2 | StructureP3 | ContentsValueP1 | ContentsValueP2 | ContentsValueP3 |
|---|---|---|---|---|---|---|
| Pinnacle Port B1 | 1924010 | 2025274 | 2126538 | 962005 | 1012637 | 1063269 |
| Pinnacle Port B8 | 776314.7 | 817173.4 | 858032.1 | 388157.4 | 408586.7 | 429016 |
| Pinnacle Port Pool | 35881.5 | 37770 | 39658.5 | 0 | 0 | 0 |
| Pinnacle Port Pool2 | 35881.5 | 37770 | 39658.5 | 0 | 0 | 0 |
| Pinnacle Port Tennis Court | 28857 | 30376 | 31894 | 0 | 0 | 0 |
| 278 Beachside | 578615.6 | 609069 | 639522.5 | 289307.8 | 304534.5 | 319761.2 |
| 278 Beachside Walkway | 10659 | 11220 | 11781 | 0 | 0 | 0 |
| 280 Beachside | 1018182 | 1071770 | 1125359 | 509090.8 | 535885 | 562679.3 |
| 280 Beachside Garage | 160265.4 | 168700.4 | 177135.4 | 80132.7 | 84350.2 | 88567.7 |

First Floor Elevation and Time to Rebuild Triangular Distribution Parameters.

| Description | First Floor Elevation P1 | First Floor Elevation P2 | First Floor Elevation P3 | Time ToRebuild1 | Time ToRebuild2 | Time ToRebuild3 |
|---|---|---|---|---|---|---|
| Pinnacle Port B1 | 9.2 | 9.8 | 10.4 | 365 | 548 | 730 |
| Pinnacle Port B8 | 27.2 | 27.8 | 28.4 | 365 | 548 | 730 |
| Pinnacle Port Pool | 6.9 | 7 | 7.1 | 365 | 548 | 730 |
| Pinnacle Port Pool2 | 6.9 | 7 | 7.1 | 365 | 548 | 730 |
| Pinnacle Port Tennis Court | 6.4 | 6.5 | 6.6 | 365 | 548 | 730 |
| 278 Beachside | 9.9 | 10.5 | 11.1 | 365 | 548 | 730 |
| 278 Beachside Walkway | 16.3 | 16.8 | 17.3 | 365 | 548 | 730 |
| 280 Beachside | 10 | 10.6 | 11.2 | 365 | 548 | 730 |
| 280 Beachside Garage | 10 | 10.6 | 11.2 | 365 | 548 | 730 |

Portion of damage function associations giving number and name (triangular distribution, erosion/structures shown here, other columns of the file show inundation/structures, flooding/structures, erosion/contents, inundations/contents, and flooding contents).

| Description | DF1 | | DF2 | | DF3 | |
|---|---|---|---|---|---|---|
| Pinnacle Port B1 | 4 | ERODP1MSTRMIN | 23 | ERODP1MSTRMOSTLIKELY | 42 | ERODP1MSTRMAX |
| Pinnacle Port B8 | 4 | ERODP1MSTRMIN | 23 | ERODP1MSTRMOSTLIKELY | 42 | ERODP1MSTRMAX |
| Pinnacle Port Pool | 7 | EROSHLSTRMIN | 26 | EROSHLSTRMOSTLIKELY | 45 | EROSHLSTRMAX |
| Pinnacle Port Pool2 | 7 | EROSHLSTRMIN | 26 | EROSHLSTRMOSTLIKELY | 45 | EROSHLSTRMAX |
| Pinnacle Port Tennis Court | 7 | EROSHLSTRMIN | 26 | EROSHLSTRMOSTLIKELY | 45 | EROSHLSTRMAX |
| 278 Beachside | 5 | ERODP1SSTRMIN | 24 | ERODP1SSTRMOSTLIKELY | 43 | ERODP1SSTRMAX |
| 278 Beachside Walkway | 6 | EROPILESTRMIN | 25 | EROPILESTRMOSTLIKELY | 44 | EROPILESTRMAX |
| 280 Beachside | 5 | ERODP1SSTRMIN | 24 | ERODP1SSTRMOSTLIKELY | 43 | ERODP1SSTRMAX |
| 280 Beachside Garage | 8 | 1SNBCMIN | 27 | 1SNBCMOSTLIKELY | 46 | 1SNBCMAX |

***DamageElementErrors.csv***

This output file contains Excel format exception report information on Damage Element errors determined in the coordinate checking process. It

is a companion to _CoordinateChecks.txt, and contains similar information, reformatted. The file is in two sections. The first section identifies Damage Elements that, by virtue of the location of their identifying point, are not in the Reach and/or the Lot to which they have been assigned. The second section evaluates the location of the Damage Element along the SBEACH line, and compares the input first floor elevation (minimum) to the ground elevation at that location. An elevation error is noted if the minimum first floor elevation (MinFFElev) is below the calculated profile elevation at that point. The location on the cross-shore morphology (e.g., S Dune Slope, seaward dune slope, etc.) is noted. Possible entries are: Beyond Foreshore, Dune, L Dune Slope (Landward Dune Slope), S Dune Slope (seaward dune slope), Dune, Upland, Negative (landward of the location of the SBEACH zero point), and Berm.

| Reach | Lot ID | Lot External ID | Damage Element ID | Description | External ID | In Reach Lot |
|---|---|---|---|---|---|---|
| R-3 | 64 | 81 | 203 | 201 Dunecrest Ln | 196 | Not In Reach |
| R-3 | 68 | 94 | 265 | 110 Village Way Garage | 265 | Not In Reach |
| R-3 | 68 | 94 | 266 | 114 Village Way | 266 | Not In Reach |
| R-3 | 68 | 94 | 267 | 114 Village Way Garage | 267 | Not In Reach |

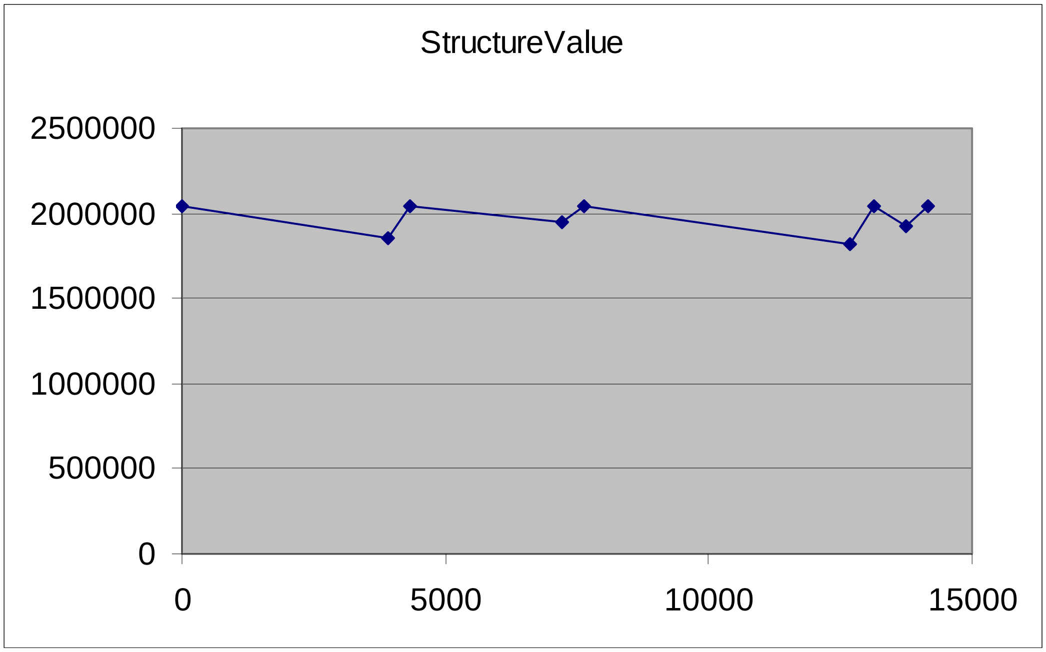
| Reach | Lot ID | Lot External ID | Damage Element ID | Description | External ID | In Reach Lot | Profile Eleve | Min FF Elev | Elevation Error | Position Error | SBEACH Position |
|---|---|---|---|---|---|---|---|---|---|---|---|
| R-1 | 1 | 1 | 2 | Pinnacle Port Gazebo | 11 | Y | 6 | 12.83 | | NEGATIVE | -55.292 |
| R-1 | 1 | 1 | 14 | Pinnacle Port Tennis Court | 14 | Y | 6 | 6.4 | | NEGATIVE | -27.028 |
| R-2 | 2 | 2 | 15 | 278 Beachside | 15 | Y | 16.5 | 9.91 | Elevation Error | DUNE | 309.341 |
| R-2 | 3 | 3 | 17 | 280 Beachside | 17 | Y | 16.5 | 10 | Elevation Error | DUNE | 311.037 |
| R-2 | 4 | 4 | 19 | 282 Beachside | 19 | Y | 16.5 | 12.19 | Elevation Error | DUNE | 312.627 |
| R-2 | 8 | 8 | 28 | 292 Beachside | 27 | Y | 16.5 | 14.42 | Elevation Error | DUNE | 311.631 |

***DamageValueHistory.csv***

This output file provides information on initial value, and all subsequent damaging and rebuilding events, as a time line, for all Damage Elements for all iterations. As such, it can get large when many iterations are run. This output file can be used to trace the value over time for individual Damage Elements, by filtering on the individual Damage Element, as shown here (only selected columns are shown).

| Iteration | Time | Description | Event | Storm ID | Remaining Rebuilds | Structure Value | Contents Value | Time To Rebuild |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | Pinnacle Port B6 | INIT | 0 | 1000 | 2041596 | 1020798 | -1 |
| 1 | 3900.959 | Pinnacle Port B6 | Storm | 385 | 1000 | 1857556 | 615261.6 | 442.597 |
| 1 | 4343.556 | Pinnacle Port B6 | Rebuild | 385 | 999 | 2041596 | 1020798 | 442.597 |
| 1 | 7205.277 | Pinnacle Port B6 | Storm | 395 | 999 | 1946451 | 598430.3 | 420.793 |
| 1 | 7626.07 | Pinnacle Port B6 | Rebuild | 395 | 998 | 2041596 | 1020798 | 420.793 |
| 1 | 12693.07 | Pinnacle Port B6 | Storm | 503 | 998 | 1823530 | 587731.8 | 449.466 |
| 1 | 13142.54 | Pinnacle Port B6 | Rebuild | 503 | 997 | 2041596 | 1020798 | 449.466 |
| 1 | 13735.82 | Pinnacle Port B6 | Storm | 339 | 997 | 1920303 | 658769.2 | 427.995 |
| 1 | 14163.81 | Pinnacle Port B6 | Rebuild | 339 | 996 | 2041596 | 1020798 | 427.995 |

This information can be plotted to show the pattern of damage and rebuilding for an individual structure:




***DECondemnation.csv***

This output file contains information on each Damage Element that is condemned. Condemnation takes place when the ratio of the structure value to the initial value is less than the Damage Element Type-specific user input condemnation ratio.

| Iteration | Time | Reach | Lot ID | Lot Description | DE | DE External ID | DE_ID | Current Value Structure | Initial Value Structure | Type | Condemnation Ratio |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3900.959 | R-2 | 3 | 3 | 280 Beachside Garage | 18 | 18 | 81229.05 | 169065.4 | SFR1 | 0.5 |
| 1 | 3900.959 | R-2 | 4 | 4 | 282 Beachside Garage | 20 | 20 | 102625 | 208598 | SFR1 | 0.5 |
| 1 | 3900.959 | R-2 | 8 | 8 | 292 Beachside Garage | 28 | 27 | 159876.9 | 338268.4 | SFR1 | 0.5 |
| 1 | 3900.959 | R-2 | 16 | 16 | 310 Beachside Garage | 43 | 43 | 85200.81 | 175717.3 | SFR1 | 0.5 |
| 1 | 3900.959 | R-2 | 22 | 22 | 322 Beachside Garage | 53 | 52 | 118897 | 243414.9 | SFR1 | 0.5 |
| 1 | 3900.959 | R-2 | 33 | 87 | 124 Seahill Ave Garage | 229 | 232 | 83381.75 | 171777.2 | SFR1 | 0.5 |
| 1 | 3900.959 | R-2 | 34 | 88 | 200 Bellview Dr Garage | 238 | 239 | 37355.16 | 75789.97 | SFR1 | 0.5 |

***DESummary1.csv***

This output file provides a rollup summary of input information on contents and value for Damage Elements, rolled up by Damage Element Type. The total number of Damage Elements is shown (Count of Structure Value P2), together with groups of Sum/Minimum/Average/Maximum values for the minimum (p1), most likely (p2), and maximum (p3) values of the triangular distribution for structure contents and value. Only the output for structure value, P1 is shown, but the actual output file contains four additional columns for each of Structure P2, Structure P3, Contents P1, Contents P2, and Contents P3. The purpose of this output file is to assist the user in determining whether any excessively large or small values may be contained in the database.

| Damage Element Type Code | Damage Element Type Description | Count of Structure Value, P2 | Sum of Structure Value, P1 | Min of Structure Value, P1 | Avg of Structure Value, P1 | Max of Structure Value, P1 |
|---|---|---|---|---|---|---|
| COMM1 | Commercial | 1 | 179105.1 | 179105.1 | 179105.1 | 179105.1 |
| GAZEBO | Gazebo | 1 | 950 | 950 | 950 | 950 |
| MFR1 | Multi-Family Residential | 9 | 13096690 | 313047.4 | 1455188 | 1924010 |
| POOL | Pool | 8 | 236407.5 | 19000 | 29550.94 | 35881.5 |
| SFR1 | Single-Family Residential | 216 | 87389704 | 68259.4 | 404582 | 1615000 |
| TENNIS | Tennis court | 1 | 28857 | 28857 | 28857 | 28857 |
| WALK | Walkway | 37 | 543452.3 | 7315 | 14687.9 | 58781.25 |

***DESummary2.csv.*** This output file is identical in character to DESummary1.csv, except that rollups are done to the Reach level. A partial extract, showing the Structure P1 values for the first two Reaches, is shown here.

| Reach Number | Reach Description | Economic Reach Number | Damage Element Type Code | Damage Element Type Description | Count of Structure Value, P2 | Sum of Structure Value, P1 | Min of Structure Value, P1 | Avg of Structure Value, P1 | Max of Structure Value, P1 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | R-1 | 1 | COMM1 | Commercial | 1 | 179105.1 | 179105.1 | 179105.1 | 179105.1 |
| 1 | R-1 | 1 | GAZEBO | Gazebo | 1 | 950 | 950 | 950 | 950 |
| 1 | R-1 | 1 | MFR1 | Multi-Family Residential | 9 | 13096690 | 313047.4 | 1455188 | 1924010 |
| 1 | R-1 | 1 | POOL | Pool | 2 | 71763 | 35881.5 | 35881.5 | 35881.5 |
| 1 | R-1 | 1 | TENNIS | Tennis court | 1 | 28857 | 28857 | 28857 | 28857 |
| 2 | R-2 | 1 | POOL | Pool | 4 | 92881.5 | 19000 | 23220.38 | 35881.5 |
| 2 | R-2 | 1 | SFR1 | Single-Family Residential | 117 | 43733997 | 68259.4 | 373794.8 | 1073500 |
| 2 | R-2 | 1 | WALK | Walkway | 16 | 234654.8 | 7315 | 14665.92 | 49637.5 |

***EmergencyNourishment.csv***

This output file provides detailed information used to verify the volumetric calculations involved in Emergency Nourishment placement. Selected columns are shown here.

| Iteration | Time | Reach | DE | BE | BW | DW | Fill Density | Scarp Distance | EN_BW Change | EN_DW Change | Nourishment Volume | Nourishment Duration | Post EN_DW | Post EN_BW |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1756 | R-1 | 8.389 | 6.6 | 36.094 | 3.344 | 6 | 0 | 0 | 38.632 | 9600 | 6.4 | 41.976 | 0 |
| 3 | 1805 | R-1 | 7.313 | 6.6 | 141.505 | 20.928 | 6 | 0 | 0 | 143.141 | 9600 | 6.4 | 164.07 | 0 |
| 4 | 622 | R-1 | 6.6 | 6.6 | 97.408 | 0 | 6 | 0 | 0 | 0 | 9600 | 6.4 | 4.426 | 97.408 |
| 4 | 633.313 | R-1 | 6.6 | 6.6 | 137.891 | 0 | 6 | 0 | 0 | 0 | 9600 | 6.4 | 4.426 | 137.891 |
| 4 | 639.713 | R-2 | 16.754 | 6.6 | -3.691 | 27.148 | 6 | 3.691 | 0 | 0 | 7200 | 4.8 | 27.708 | 0 |
| 4 | 644.513 | R-3 | 16.754 | 6.6 | -2.292 | 27.148 | 6 | 2.292 | 0 | 0 | 6000 | 4 | 28.813 | 0 |
| 4 | 648.513 | R-4 | 16.754 | 6.6 | -0.636 | 27.148 | 6 | 0.636 | 0 | 0 | 6270 | 4.18 | 30.115 | 0 |
| 5 | 1455 | R-1 | 7.5 | 6.6 | 30.006 | 33.571 | 6 | 0 | 0 | 33.605 | 9600 | 6.4 | 67.176 | 0 |

***ENSummary.csv***

This output file provides a statistical summary of the number of Emergency Nourishments, volume, and cost, by Reach, over all iterations.

| Reach | Average Number | Max Number | Min Number | SD Number | Average Volume | Max Volume | Min Volume | SD Volume | Average Cost | Max Cost | Min cost | SD Cost |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R-1 | 0.8 | 1 | 0 | 0.4 | 7680 | 9600 | 0 | 4293.3 | 244510.3 | 340155.5 | 0 | 138242.6 |
| R-2 | 0.2 | 1 | 0 | 0.4 | 1440 | 7200 | 0 | 3219.9 | 50989 | 254945 | 0 | 114014.9 |
| R-3 | 0.2 | 1 | 0 | 0.4 | 1200 | 6000 | 0 | 2683.3 | 42467 | 212335.1 | 0 | 94959.1 |
| R-4 | 0.2 | 1 | 0 | 0.4 | 1254 | 6270 | 0 | 2804 | 44352 | 221760.2 | 0 | 99174.2 |

### *Iteration.csv*

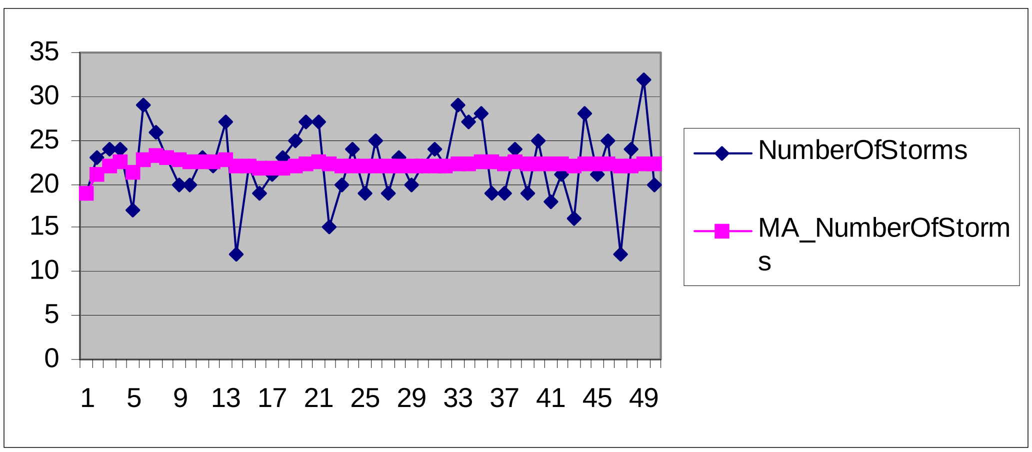
This output file presents summary information on each iteration, with information as follows:

| Iteration | Iteration Number |
|---|---|
| NumberOfStorms | Number of storms in iteration |
| MA_NumberOfStorms | Moving Average over this and prior iterations of number of storms |
| EmergencyNourishmentCost | Present value total Emergency Nourishment cost in iteration |
| MA_EmergencyNourishmentCost | Moving average present value Emergency Nourishment cost over this and prior iterations |
| EmergencyNourishmentMobilizationCost | Present value total Emergency Nourishment mobilization cost in iteration |
| MA_EmergencyNourishmentMobilizationCost | Moving average present value Emergency Nourishment mobilization cost over this and prior iterations |
| PlannedMobilizationCost | Present value total planned mobilization cost in iteration |
| MA_PlannedMobilizationCost | Moving average present value planned mobilization cost |
| PVPlannedPlacementCost | Present value of planned placement cost in iteration |
| MA_PVPlannedPlacementCost | Moving average of present value of planned placement cost |
| PVDamagesCost | Present value total damages cost in iteration |
| MA_PVDamagesCost: | Moving average of total damages cost |

Results from a five-iteration run (selected columns) are shown here:

| Iteration | Number of Storms | MA Number of Storms | Emergency Nourishment Cost | MA Emergency Nourishment Cost | PV Planned Placement Cost | MA PV Planned Placement Cost | PV Damages Cost | MA PV Damages Cost |
|---|---|---|---|---|---|---|---|---|
| 1 | 19 | 19 | 0 | 0 | 8510745 | 8510745 | 6653516 | 6653516 |
| 2 | 23 | 21 | 290620 | 145309.999 | 7908125 | 8209435 | 6977128 | 6815322 |
| 3 | 24 | 22 | 288630.5 | 193083.507 | 7313344 | 7910738 | 6904355 | 6845000 |
| 4 | 24 | 22.5 | 1029196 | 402111.555 | 7636213 | 7842107 | 18718350 | 9813337 |
| 5 | 17 | 21.4 | 303145.5 | 382318.344 | 7176923 | 7709070 | 0 | 7850670 |

It can be used to easily create graphical output (example taken from 50 iteration run), e.g.:



***LandLoss.csv***

This output file reports end of year values and statistics calculated throughout the year for each Reach. The end of the year is 1 year after the start simulation date, in this case the start date of the simulation is 1 January 2000. Berm section is calculated as the distance from seaward toe of the dune to start of fore slope. Dune section is the distance from the seaward toe of the dune to the landward toe of the dune. Upland width is the distance between the SBEACH cross-shore position and the landward toe of the dune. It can be used for offline calculations for land loss and recreation benefits values.

| Iteration | Time | Date | Reach | Economic Reach | Description | Berm Section | Dune Section | Upland Width Section |
|---|---|---|---|---|---|---|---|---|
| 1 | 366 | 1/1/2001 | 1 | 1 | R-1 | 1.66 | 108.333 | 300 |
| 1 | 731 | 1/1/2002 | 1 | 1 | R-1 | 3.321 | 108.333 | 300 |
| 1 | 1096 | 1/1/2003 | 1 | 1 | R-1 | 4.981 | 108.333 | 300 |
| 1 | 1461 | 1/1/2004 | 1 | 1 | R-1 | 6.642 | 108.333 | 300 |
| 1 | 1827 | 1/1/2005 | 1 | 1 | R-1 | 8.302 | 108.333 | 300 |
| 1 | 2192 | 1/1/2006 | 1 | 1 | R-1 | 92.198 | 134.556 | 300 |
| 1 | 2557 | 1/1/2007 | 1 | 1 | R-1 | 86.878 | 134.556 | 300 |
| 1 | 2922 | 1/1/2008 | 1 | 1 | R-1 | 81.557 | 134.556 | 300 |
| 1 | 3288 | 1/1/2009 | 1 | 1 | R-1 | 76.237 | 134.556 | 300 |
| 1 | 3653 | 1/1/2010 | 1 | 1 | R-1 | 66.691 | 134.556 | 300 |

| Time | Date | Berm Average | Berm Min | Berm Max | Berm SD | Dune Average | Dune Min | Dune Max | Dune SD | Upland Average | Upland Min | Upland Max | Upland SD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 366 | 1/1/2001 | 0.846 | 0.032 | 1.66 | 0.484 | 108.333 | 108.333 | 108.333 | 0 | 300 | 300 | 300 | 0 |
| 731 | 1/1/2002 | 2.507 | 1.692 | 3.321 | 0.484 | 108.333 | 108.333 | 108.333 | 0 | 300 | 300 | 300 | 0 |
| 1096 | 1/1/2003 | 4.167 | 3.353 | 4.981 | 0.484 | 108.333 | 108.333 | 108.333 | 0 | 300 | 300 | 300 | 0 |
| 1461 | 1/1/2004 | 5.827 | 5.013 | 6.642 | 0.484 | 108.333 | 108.333 | 108.333 | 0 | 300 | 300 | 300 | 0 |
| 1827 | 1/1/2005 | 7.488 | 6.674 | 8.302 | 0.484 | 108.333 | 108.333 | 108.333 | 0 | 300 | 300 | 300 | 0 |
| 2192 | 1/1/2006 | 41.392 | 8.334 | 100.128 | 41.506 | 118.723 | 108.333 | 134.556 | 12.948 | 300 | 300 | 300 | 0 |
| 2557 | 1/1/2007 | 89.487 | 86.878 | 92.096 | 1.551 | 134.556 | 134.556 | 134.556 | 0 | 300 | 300 | 300 | 0 |
| 2922 | 1/1/2008 | 84.166 | 81.557 | 86.776 | 1.551 | 134.556 | 134.556 | 134.556 | 0 | 300 | 300 | 300 | 0 |
| 3288 | 1/1/2009 | 78.846 | 76.237 | 81.455 | 1.551 | 134.556 | 134.556 | 134.556 | 0 | 300 | 300 | 300 | 0 |
| 3653 | 1/1/2010 | 70.525 | 39.866 | 76.135 | 6.537 | 134.556 | 134.556 | 134.556 | 0 | 300 | 300 | 300 | 0 |

### *LookupQuality.csv*

This output file reports out situations in which the lookup into the SRD falls outside of parameters set in tblConfigurationSettings for tolerance (values in feet). Note that this is an exception report, only lookups outside the tolerances are reported.

| Configuration Setting ID | Tag | Description | Value |
|---|---|---|---|
| 9 | LookupBermWidthExceptionReportTolerance | Value in feet for flagging exception report on lookups for berm width | 20 |
| 10 | LookupDuneWidthExceptionReportTolerance | Value in feet for flagging exception report on lookups for dune width | 20 |
| 11 | LookupDuneHeightExceptionReportTolerance | Value in feet for flagging exception report on lookups for dune height | 20 |

When the best lookup is found, the return values are tested against these tolerances, if one or more of the returned values for berm width, dune width, or dune height differ from the respective lookup inputs (pre-storm berm width, dune width, dune height) by the tolerance amount, the lookup is reported to this file. “Code” is a binary code, 1 = berm width, 2 = dune width, 4 = dune height, and combinations, where two or more are exceeded, are reported as the sum (e.g., a code of 3 means that both berm width and dune width lookup tolerances are exceeded).

| Iteration | Time | Reach | Storm ID | Storm Type | Storm Response ID | Pre Storm Berm Width | Pre Storm Dune Width | Pre Storm Dune Height | Lookup Berm Width | Lookup Dune Width | Lookup Dune Height | Berm Width Difference | Dune Width Difference | Dune Height Difference | Code |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4569.5 | R-1 | 81 | 1 | 40686 | 48.628 | 15.95 | 8.096 | 50 | 95 | 8.5 | 1.372 | 79.05 | 0.404 | 2 |
| 1 | 13038.5 | R-1 | 526 | 1 | 46387 | 77.135 | 22.393 | 8.026 | 80 | 95 | 8.5 | 2.865 | 72.607 | 0.474 | 2 |
| 1 | 13735.8 | R-1 | 339 | 1 | 27544 | 68.096 | 21.238 | 8.026 | 70 | 95 | 8.5 | 1.904 | 73.762 | 0.474 | 2 |

### *LotCondemnation.csv*

This output file is used to record the Lot condemnation process. It reports each Lot condemnation, based on the centroid of the Lot being within a user-specified (at Reach level) distance from the seaward toe of the dune. The file reports the centroid location on the SBEACH Line, the current seaward toe of the dune (reported as shoreline location), the calculated critical location (seaward toe of the dune location -control line offset), and the difference between the centroid location and the seaward toe of the dune location, which should, in every case, be greater than (-1 * control line offset). A positive value of the difference means that the calculated seaward toe of the dune is landward of the Lot centroid, while a negative difference means that the seaward toe of the dune is seaward of the centroid. The file also reports current and initial values of structure and contents on the Lot (only the structure columns are shown in the following example).

| Iteration | Time | Reach | Lot ID | Location on SBEACH Line | Shoreline Location | Control Line Offset | Critical Distance | Difference | Current Upland Width | Initial Value Structure | Current Value Structure |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 12693.07 | R-2 | 2 | 376.749 | 366.61 | 0 | 366.61 | 10.138 | 205 | 628151.5 | 628151.5 |
| 1 | 12693.07 | R-2 | 3 | 378.355 | 366.61 | 0 | 366.61 | 11.745 | 205 | 1244743 | 1244743 |
| 1 | 12693.07 | R-2 | 4 | 379.163 | 366.61 | 0 | 366.61 | 12.553 | 205 | 829148.1 | 829148.1 |
| 1 | 12693.07 | R-2 | 5 | 380.303 | 366.61 | 0 | 366.61 | 13.692 | 205 | 558136.7 | 558136.7 |
| 1 | 12693.07 | R-2 | 6 | 377.184 | 366.61 | 0 | 366.61 | 10.573 | 205 | 106226.1 | 106226.1 |
| 1 | 12693.07 | R-2 | 7 | 374.666 | 366.61 | 0 | 366.61 | 8.056 | 205 | 1026172 | 1026172 |
| 1 | 13735.82 | R-2 | 2 | 376.749 | 343.427 | 0 | 343.427 | 33.322 | 205 | 628151.5 | 628151.5 |
| 1 | 13735.82 | R-2 | 3 | 378.355 | 343.427 | 0 | 343.427 | 34.928 | 205 | 1244743 | 1244743 |
| 1 | 13735.82 | R-2 | 4 | 379.163 | 343.427 | 0 | 343.427 | 35.736 | 205 | 829148.1 | 829148.1 |
| 1 | 13735.82 | R-2 | 5 | 380.303 | 343.427 | 0 | 343.427 | 36.875 | 205 | 558136.7 | 558136.7 |

### *LotEcho.csv*

This output file echoes the coordinates of the Lot quadrilateral, and provides the calculated location (state plane coordinates) of the Lot centroid, as well as the location of the centroid on the SBEACH line, and the location of the closest quadrilateral point to the shore, as measured on the SBEACH line. This output file also contains information about Lot armoring, not shown in the following examples.

| Lot ID | Reach Number | Lot External ID | Lot Description | P1x | P1y | P2x | P2y | P3x | P3y | P4x | P4y |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1498619 | 463902.7 | 1499956 | 463870.1 | 1499881 | 462871.4 | 1498408 | 463655.9 |
| 2 | 2 | 2 | 2 | 1500026 | 463168.4 | 1500049 | 463144.2 | 1499909 | 462859.5 | 1499881 | 462871.4 |
| 3 | 2 | 3 | 3 | 1500049 | 463144.2 | 1500077 | 463123.7 | 1499943 | 462852 | 1499909 | 462859.5 |
| 4 | 2 | 4 | 4 | 1500077 | 463123.7 | 1500104 | 463106.5 | 1499971 | 462834.7 | 1499943 | 462852 |

| Lot ID | Centroid_x | Centroid_y | Centroid_Location_SBeach | ClosestShorePoint_Location_SBeach |
|---|---|---|---|---|
| 1 | 1499216 | 463575 | 130.362 | 455.393 |
| 2 | 1499966 | 463010.9 | 376.749 | 539.745 |
| 3 | 1499995 | 462994.8 | 378.355 | 537.677 |
| 4 | 1500024 | 462979.2 | 379.163 | 532.064 |

### *LotErrors.csv*

This output file provides a variety of messages about error checks for the Lot. The centroid and corner points are tested as to whether or not they are in the Reach quadrilateral, and the Lot area is flagged if it is too large or small (based on Configuration Settings, currently set to between 0.1 and 5 acres, but user-definable) – this is a warning, not necessarily an error:

| Reach | Lot ID | Lot | Area | Message |
|---|---|---|---|---|
| R-1 | 1 | 1 | 21.42 | Lot Area |
| R-2 | 2 | 2 | 0.23 | Lot Area |
| R-2 | 3 | 3 | 0.24 | Lot Area |
| R-2 | 4 | 4 | 0.22 | Lot Area |
| R-2 | 5 | 5 | 0.34 | Lot Area |
| R-2 | 6 | 6 | 0.7 | Lot Area |
| R-2 | 7 | 7 | 0.31 | Lot Area |

In addition, the corners and centroid of each Lot are checked against the Reach it is supposed to be in, and other Reaches, in a somewhat cryptic output format at present, where a 1 indicates that the corner is in the Reach.

| Reach | Lot | | Area | Centroid | P1 | P2 | P3 | P4 | Total | Other Reach quadri-lateral | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R-1 | 1 | 1 | 0.58 | 1 | 1 | 1 | 0 | 1 | 4 | R-1 | Centroid in Reach | Other Points Not |
| R-2 | 1 | 1 | 0.58 | 0 | 0 | 0 | 1 | 0 | 1 | R-1 | Corners in Different Reach | |
| R-2 | 22 | 22 | 0.58 | 1 | 1 | 1 | 0 | 1 | 4 | R-2 | Centroid in Reach | Other Points Not |
| R-2 | 30 | 80 | 0.58 | 1 | 1 | 1 | 0 | 1 | 4 | R-2 | Centroid in Reach | Other Points Not |
| R-2 | 39 | 93 | 0.58 | 1 | 1 | 0 | 1 | 1 | 4 | R-2 | Centroid in Reach | Other Points Not |
| R-2 | 42 | 23 | 0.58 | 0 | 1 | 0 | 0 | 0 | 1 | R-3 | Corners in Different Reach | |
| R-2 | 64 | 81 | 0.58 | 0 | 1 | 0 | 0 | 0 | 1 | R-3 | Corners in Different Reach | |
| R-2 | 68 | 94 | 0.58 | 0 | 0 | 0 | 0 | 1 | 1 | R-3 | Corners in Different Reach | |

### *Memory.csv*

The memory.csv file is used to check the memory usage as the simulation proceeds, and is considered a debugging file, not of interest to the user. It reports memory usage at the end of each iteration.

### *Message.csv*

This output file provides messages from the simulation. Messages include start and end time of the run, warnings, and critical messages. If a critical message is written, then, at the end of the simulation, the user is advised of that, such that the message.csv file can be examined, and, if needed, information can be transmitted to the support team.

### *MorphologyConstants.csv*

This output file provides Profile-based Reach constants for use in post-processing animation.

| Reach | Profile | Upland Elevation | Dune Slope | Berm Elevation | Foreshore Slope | Historic Rate |
|---|---|---|---|---|---|---|
| R-1 | R1 | 6 | 0.18 | 6.6 | 0.0588 | 1.665 |
| R-2 | R2R5 | 6 | 0.106 | 6.6 | 0.0588 | -0.765 |
| R-3 | R2R5 | 6 | 0.106 | 6.6 | 0.0588 | 0.036 |
| R-4 | R2R5 | 6 | 0.106 | 6.6 | 0.0588 | 0.975 |

### *MorphologyTimeLine.csv*

This output file provides information on all significant morphology change events taking place over time. It is used in post-processing animation. The following is a portion of the file, filtered to show only Reach R1. The final column reports additional information, depending upon the event type.

| Iteration | Time | Reach | Upland Width | Dune Height | Dune Width | Berm Width | Event Type | |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | R-1 | 300 | 7.5 | 95 | 0 | INIT | NA |
| 1 | 2039.02 | R-1 | 300 | 7.5 | 95 | 9.292 | SSN | NA |
| 1 | 2047.955 | R-1 | 300 | 9.5 | 99 | 100 | ESN | NA |
| 1 | 2104.949 | R-1 | 300 | 9.5 | 99 | 99.718 | Pre Storm | WCe_19641004_H2 |
| 1 | 2105.949 | R-1 | 300 | 9.5 | 99 | 37.817 | Post Storm | WCe_19641004_H2 |
| 1 | 2125.949 | R-1 | 300 | 9.5 | 99 | 93.221 | REC | WCe_19641004_H2 |
| 1 | 3490.871 | R-1 | 300 | 9.5 | 99 | 73.27 | Pre Storm | WCe_18870727_H2 |
| 1 | 3491.871 | R-1 | 300 | 9.5 | 99 | 36.625 | Post Storm | WCe_18870727_H2 |
| 1 | 3511.871 | R-1 | 300 | 9.5 | 99 | 69.298 | REC | WCe_18870727_H2 |

### *Nourishment.csv*

Information is provided for each nourishment (E = Emergency, P = Planned) event for each Reach. “Number” is the number of Emergency Nourishments for the Reach in the iteration.

| Iteration | Start Time | Start Date | End Time | End Date | Duration | Reach | Length | Type | Economic Reach | Unit Placement Cost |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2039.02 | 8/1/2005 | 2047.955 | 8/9/2005 | 8.935 | R-1 | 1600 | P | 1 | 5 |
| 1 | 2047.965 | 8/9/2005 | 2057.129 | 8/19/2005 | 9.163 | R-2 | 1200 | P | 1 | 5 |
| 1 | 2057.139 | 8/19/2005 | 2064.571 | 8/26/2005 | 7.432 | R-3 | 1000 | P | 1 | 5 |
| 1 | 2064.581 | 8/26/2005 | 2072.097 | 9/3/2005 | 7.517 | R-4 | 1045 | P | 1 | 5 |
| 1 | 4594.02 | 7/30/2012 | 4605.79 | 8/10/2012 | 11.77 | R-1 | 1600 | P | 1 | 5 |
| 1 | 4605.8 | 8/10/2012 | 4609.494 | 8/14/2012 | 3.694 | R-2 | 1200 | P | 1 | 5 |
| 1 | 4609.504 | 8/14/2012 | 4612.46 | 8/17/2012 | 2.957 | R-3 | 1000 | P | 1 | 5 |
| 1 | 4612.47 | 8/17/2012 | 4615.478 | 8/20/2012 | 3.008 | R-4 | 1045 | P | 1 | 5 |

| Reach | Total Nourishment Volume cu yds | Nourishment cu yd/ft | Upper Beach Nourishment Volume cu yd/ft | Lower Beach Nourishment Volume cu yd/ft | PV Mobilization | PV Placement Cost | Number | Total Cost |
|---|---|---|---|---|---|---|---|---|
| R-1 | 268057.2 | 167.536 | 0 | 0 | 0 | 1299407 | 0 | 1299407 |
| R-2 | 274898.7 | 229.082 | 0 | 0 | 0 | 1330858 | 0 | 2630265 |
| R-3 | 222964.8 | 222.965 | 0 | 0 | 0 | 1078307 | 0 | 3708572 |
| R-4 | 225497.4 | 215.787 | 0 | 0 | 0 | 1089405 | 0 | 4797977 |
| R-1 | 353088 | 220.68 | 0 | 0 | 0 | 1195841 | 0 | 5993818 |
| R-2 | 110819.2 | 92.349 | 0 | 0 | 0 | 375128.2 | 0 | 6368946 |
| R-3 | 88695.06 | 88.695 | 0 | 0 | 0 | 300112.2 | 0 | 6669058 |
| R-4 | 90232.31 | 86.347 | 0 | 0 | 0 | 305184.6 | 0 | 6974243 |

| Reach | Pre Nourishment Dune Height | Pre Nourishment Dune Width | Pre Nourishment Berm Width | Template Dune Height | Template Dune Width | Template Berm Width | Trigger Dune Height | Trigger Dune Width | Trigger Berm Width |
|---|---|---|---|---|---|---|---|---|---|
| R-1 | 7.5 | 95 | 9.324 | 9.5 | 99 | 100 | 0.8 | 0.6 | 0.5 |
| R-2 | 16.5 | 45 | -4.299 | 18.5 | 55 | 100 | 0.9 | 0.6 | 0.5 |
| R-3 | 16.5 | 45 | 0.203 | 18.5 | 55 | 100 | 0.9 | 0.6 | 0.5 |
| R-4 | 16.5 | 45 | 5.535 | 18.5 | 55 | 100 | 0.9 | 0.6 | 0.5 |
| R-1 | 8.096 | 15.775 | 44.064 | 9.5 | 99 | 100 | 0.8 | 0.6 | 0.5 |
| R-2 | 18.5 | 55 | 31.74 | 18.5 | 55 | 100 | 0.9 | 0.6 | 0.5 |
| R-3 | 18.5 | 55 | 34.569 | 18.5 | 55 | 100 | 0.9 | 0.6 | 0.5 |
| R-4 | 18.5 | 55 | 36.182 | 18.5 | 55 | 100 | 0.9 | 0.6 | 0.5 |

***NourishmentCalendar.csv***

The nourishment calendar output file provides nourishment timing information by date, in order to verify the behavior with respect to user input nourishment blackout windows. The first blackout window is assigned to the year prior to the simulation start year, to ensure that over-year blackout window specifications are properly captured at the start of the simulation. Nourishments are allowed to start 0.1 days after the end of the blackout window, but no portion of the projected nourishment duration can overlap with any portion of the blackout window. Note: this example will not be consistent in terms of nourishment timing with other nourishment examples provided, which have been generated from simulations without specified blackout windows.

| Iteration | Type | Reach | Start Time | End Time | Start Date | End Date |
|---|---|---|---|---|---|---|
| 1 | BW | - | -153 | -139 | 8/1/1999 0:00 | 8/15/1999 0:00 |
| 1 | BW | - | 213 | 227 | 8/1/2000 0:00 | 8/15/2000 0:00 |
| 1 | BW | - | 578 | 592 | 8/1/2001 0:00 | 8/15/2001 0:00 |
| 1 | BW | - | 943 | 957 | 8/1/2002 0:00 | 8/15/2002 0:00 |
| 1 | BW | - | 1308 | 1322 | 8/1/2003 0:00 | 8/15/2003 0:00 |
| 1 | BW | - | 1674 | 1688 | 8/1/2004 0:00 | 8/15/2004 0:00 |
| 1 | BW | - | 2039 | 2053 | 8/1/2005 0:00 | 8/15/2005 0:00 |
| 1 | PN | R-1 | 2053.1 | 2062.031 | 8/15/2005 2:24 | 8/24/2005 0:44 |
| 1 | PN | R-2 | 2062.041 | 2071.206 | 8/24/2005 0:58 | 9/2/2005 4:55 |
| 1 | PN | R-3 | 2071.216 | 2078.648 | 9/2/2005 5:10 | 9/9/2005 15:32 |
| 1 | PN | R-4 | 2078.658 | 2086.172 | 9/9/2005 15:46 | 9/17/2005 4:08 |
| 1 | BW | - | 2404 | 2418 | 8/1/2006 0:00 | 8/15/2006 0:00 |
| 1 | BW | - | 2769 | 2783 | 8/1/2007 0:00 | 8/15/2007 0:00 |
| 1 | BW | - | 3135 | 3149 | 8/1/2008 0:00 | 8/15/2008 0:00 |
| 1 | BW | - | 3500 | 3514 | 8/1/2009 0:00 | 8/15/2009 0:00 |
| 1 | BW | - | 3865 | 3879 | 8/1/2010 0:00 | 8/15/2010 0:00 |
| 1 | BW | - | 4230 | 4244 | 8/1/2011 0:00 | 8/15/2011 0:00 |
| 1 | BW | - | 4596 | 4610 | 8/1/2012 0:00 | 8/15/2012 0:00 |
| 1 | PN | R-1 | 4610.1 | 4621.87 | 8/15/2012 2:24 | 8/26/2012 20:52 |
| 1 | PN | R-2 | 4621.88 | 4625.581 | 8/26/2012 21:06 | 8/30/2012 13:56 |
| 1 | PN | R-3 | 4625.591 | 4628.575 | 8/30/2012 14:10 | 9/2/2012 13:48 |
| 1 | PN | R-4 | 4628.585 | 4631.599 | 9/2/2012 14:02 | 9/5/2012 14:22 |

***PlannedNourishment.csv***

This output file provides project level information on each Planned Nourishment cycle.

| Iteration | Time | Cycle | Cycle Start Time | Cycle End Time | Mobilization CostPV | Total Volume for Mobilization | Total Volume in Nourishment | Number of Reaches Nourished in Cycle | Nourishment Cost in Cycle |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 2072.097 | 1 | 2039 | 2072.097 | 388287.1 | 991475.2 | 991418.1 | 4 | 4797977 |
| 1 | 4615.478 | 2 | 4594 | 4615.478 | 271392.7 | 641937.7 | 642834.5 | 4 | 2176266 |
| 1 | 9722.639 | 3 | 9704 | 9722.639 | 132583.3 | 557206 | 557656.7 | 4 | 922684.7 |
| 1 | 14839.39 | 4 | 14814 | 14839.39 | 64770.8 | 759385.3 | 760094.9 | 4 | 613817.6 |

***PlannedNourishmentReach.csv***

This output file presents information on each phase of scheduled nourishment for each Reach (C = checking, S = start, E = end) in each cycle. Note that this column normally appears as the final column in the file. Values can differ between checking and nourishment start, due to deferring of nourishment because of storms or blackout windows. Much of this information is used for verifying volumetric calculations involved in the Planned Nourishment.

| Iteration | Time | Reach | Cycle | Type | Pre-Nourishment Berm Width | Pre-Nourishment Dune Width | Pre-Nourishment Dune Height | Pre-Nourishment Upland Width | Berm Width Trigger | Dune Width Trigger | Dune Height Trigger |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2039 | R-1 | 1 | C | 9.292 | 95 | 7.5 | 300 | 50 | 59.4 | 7.6 |
| 1 | 2039.02 | R-1 | 1 | S | 9.292 | 95 | 7.5 | 300 | 100 | 99 | 9.5 |
| 1 | 2047.955 | R-1 | 1 | E | 9.292 | 95 | 7.5 | 300 | 100 | 99 | 9.5 |
| 1 | 4594 | R-1 | 2 | C | 44.167 | 15.775 | 8.096 | 300 | 50 | 59.4 | 7.6 |
| 1 | 4594.02 | R-1 | 2 | S | 44.167 | 15.775 | 8.096 | 300 | 100 | 99 | 9.5 |
| 1 | 4605.79 | R-1 | 2 | E | 44.167 | 15.775 | 8.096 | 300 | 100 | 99 | 9.5 |
| 1 | 9704 | R-1 | 3 | C | 155.016 | 10.953 | 8.789 | 272.948 | 50 | 59.4 | 7.6 |
| 1 | 9704.02 | R-1 | 3 | S | 155.016 | 10.953 | 8.789 | 272.948 | 100 | 99 | 9.5 |
| 1 | 9709.463 | R-1 | 3 | E | 155.016 | 10.953 | 8.789 | 272.948 | 100 | 99 | 9.5 |
| 1 | 14814 | R-1 | 4 | C | 55.612 | 0 | 6.6 | 262.196 | 50 | 59.4 | 7.6 |
| 1 | 14814.02 | R-1 | 4 | S | 55.612 | 0 | 6.6 | 262.196 | 100 | 99 | 9.5 |
| 1 | 14830.06 | R-1 | 4 | E | 55.612 | 0 | 6.6 | 262.196 | 100 | 99 | 9.5 |

Morphology Types: LU = low upland; HU = high upland, LB = low berm.

| Time | Post Nourishment Berm Width | Post Nourishment Dune Width | Post Nourishment Dune Height | Post Nourishment Upland Width | Template Morphology Type | Template Dune Volume | Template Start of Berm | Template End of Berm |
|---|---|---|---|---|---|---|---|---|
| 2039 | 100 | 99 | 9.5 | 300 | LU | 15.317 | 434.556 | 534.556 |
| 2039.02 | 100 | 99 | 9.5 | 300 | LU | 15.317 | 434.556 | 534.556 |
| 2047.955 | 100 | 99 | 9.5 | 300 | LU | 15.317 | 434.556 | 534.556 |
| 4594 | 100 | 99 | 9.5 | 300 | LU | 15.317 | 434.556 | 534.556 |
| 4594.02 | 100 | 99 | 9.5 | 300 | LU | 15.317 | 434.556 | 534.556 |
| 4605.79 | 100 | 99 | 9.5 | 300 | LU | 15.317 | 434.556 | 534.556 |
| 9704 | 100 | 99 | 9.5 | 300 | LU | 15.317 | 434.556 | 534.556 |
| 9704.02 | 100 | 99 | 9.5 | 300 | LU | 15.317 | 434.556 | 534.556 |
| 9709.463 | 100 | 99 | 9.5 | 300 | LU | 15.317 | 434.556 | 534.556 |
| 14814 | 100 | 99 | 9.5 | 300 | LU | 15.317 | 434.556 | 534.556 |
| 14814.02 | 100 | 99 | 9.5 | 300 | LU | 15.317 | 434.556 | 534.556 |
| 14830.06 | 100 | 99 | 9.5 | 300 | LU | 15.317 | 434.556 | 534.556 |

The Planned Nourishment Code is an additive code that combines the values of berm width triggered (yes = 1), dune width triggered (yes = 2), and dune height triggered (yes = 4), such that the single code number allows for interpretation of which of the three elements were triggered. For example, a code value of 3 indicates that berm width (1) and dune width (2) were both triggered. Likewise, a code value of 5 indicates that berm width (1) and dune height (4) were both triggered. A Planned Nourishment code value of 6 indicates that dune width (2) and dune height (4) were both triggered. Finally, a Planned Nourishment code value of 7 indicates that berm width (1), dune width (2) and dune height (4) all triggered.

| Time | Current Dune Volume | Current Start of Berm | Current End of Berm | Number of Planned Nourishments in Iteration | Planned Nourishment Code | Berm Width Triggered | Dune Width Triggered | Dune Height Triggered |
|---|---|---|---|---|---|---|---|---|
| 2039 | 5.704 | 408.333 | 417.625 | 0 | 5 | Y | N | Y |
| 2039.02 | 5.704 | 408.333 | 417.625 | 0 | 7 | Y | Y | Y |
| 2047.955 | 5.704 | 408.333 | 417.625 | 1 | 7 | Y | Y | Y |
| 4594 | 2.091 | 335.729 | 379.895 | 1 | 3 | Y | Y | N |
| 4594.02 | 2.091 | 335.729 | 379.895 | 1 | 7 | Y | Y | Y |
| 4605.79 | 2.091 | 335.729 | 379.895 | 2 | 7 | Y | Y | Y |
| 9704 | 2.695 | 311.554 | 466.57 | 2 | 2 | N | Y | N |
| 9704.02 | 2.695 | 311.554 | 466.57 | 2 | 6 | N | Y | Y |
| 9709.463 | 2.695 | 311.554 | 466.57 | 3 | 6 | N | Y | Y |
| 14814 | 0.037 | 265.53 | 321.142 | 3 | 6 | N | Y | Y |
| 14814.02 | 0.037 | 265.53 | 321.142 | 3 | 7 | Y | Y | Y |
| 14830.06 | 0.037 | 265.53 | 321.142 | 4 | 7 | Y | Y | Y |

| Time | Planned Nourishment Duration | Planned Nourishment Volume | Berm Nourishment Volume Per Foot | Dune Nourishment Volume Per Foot | Total Nourishment Volume in Iteration | Total Nourishment Cost PV in Iteration | Planned Nourishment Placement Cost | Planned Nourishment Placement Cost PV |
|---|---|---|---|---|---|---|---|---|
| 2039 | 8.935 | 268057.2 | 158.505 | 9.03 | 0 | 0 | 0 | 0 |
| 2039.02 | 8.935 | 268057.2 | 158.505 | 9.03 | 0 | 0 | 0 | 0 |
| 2047.955 | 8.935 | 268057.2 | 158.505 | 9.03 | 268057.2 | 1299407 | 1340286 | 1299407 |
| 4594 | 11.77 | 353088 | 209.651 | 11.029 | 268057.2 | 1299407 | 1340286 | 1299407 |
| 4594.02 | 11.77 | 353088 | 209.651 | 11.029 | 268057.2 | 1299407 | 1340286 | 1299407 |
| 4605.79 | 11.77 | 353088 | 209.651 | 11.029 | 621145.2 | 2495248 | 1765440 | 1195841 |
| 9704 | 5.443 | 163276.3 | 92.159 | 9.889 | 621145.2 | 2495248 | 1765440 | 1195841 |
| 9704.02 | 5.443 | 163276.3 | 92.159 | 9.889 | 621145.2 | 2495248 | 1765440 | 1195841 |
| 9709.463 | 5.443 | 163276.3 | 92.159 | 9.889 | 784421.5 | 2765637 | 816381.6 | 270389.2 |
| 14814 | 16.044 | 481308.5 | 289.294 | 11.524 | 784421.5 | 2765637 | 816381.6 | 270389.2 |
| 14814.02 | 16.044 | 481308.5 | 289.294 | 11.524 | 784421.5 | 2765637 | 816381.6 | 270389.2 |
| 14830.06 | 16.044 | 481308.5 | 289.294 | 11.524 | 1265730 | 3154445 | 2406543 | 388807.7 |

***ProfileResponse.dbg***

This ASCII output file contains debugging information for the morphology change analysis, and is intended for use by the development/support team. An example portion of file contents is shown here:

```
Case 13
I:    1   2104.949       R-1 SRID:       5327 Catastrophic: 0
          Scarping: N Max Scarp Exceeded: N Dynamic Scarp: N

          No Scarping Beyond Maximum, no values outside of limits

               Pre        Lookup    Adj. Lookup    Tentative  Calculated
          UW:      300.000        0.000       0.000    300.000    300.000
          DH:        9.500        0.000       0.000      9.500      9.500
          DW:       99.000        0.000       0.000     99.000     99.000
          Berm:     99.718      -61.902     -61.902     37.817     37.817
          Post Recovery Berm Width:     93.528
```

***Reach.csv***

The Reach file reports information for each Reach, at user-specified multiples of the time-step. For example, if the time-step is 7 days, the user

can specify values of 0 (no output), 1 (output every 7 days), 2 (output every 14 days, or every 2nd time-step), etc. The multiplier specified in the Configuration Settings branch of the *Navigation Tree* for tag TimeStepReportInterval.

| ConfigurationSettingID | Tag | Description | Value |
|---|---|---|---|
| 6 | TimeStepReportInterval | Report every nth time step (0 = no report, 1 = every time step, 2= every 2nd time step, etc.) | 10 |

The file reports the value at the time-step, and statistical values of berm width, dune width, dune height, and upland width, calculated from one report period to the next, in this case every 10 weeks (assuming 7 day time-step). NOTE: This Output Option should be used with caution, as this file can grow very large. In the example, a nourishment event has taken place between 7/23/2005 and 10/1/2005, leading to restoration of dune height, dune width, and berm width to the template, but the berm width has eroded from the template value of 100 between the time of placement and the time of reporting (every 10 weeks).

The seaward berm location is the cross-shore location on the SBEACH line.

| Iteration | Time | Date | Reach | Economic Reach | Description | Dune Height | Dune Width | Berm Width | Upland Width | Seaward Berm |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1820 | 12/5/2004 | 1 | 1 | R-1 | 7.5 | 95 | 8.302 | 300 | 416.636 |
| 1 | 1890 | 3/5/2005 | 1 | 1 | R-1 | 7.5 | 95 | 8.622 | 300 | 416.955 |
| 1 | 1960 | 5/14/2005 | 1 | 1 | R-1 | 7.5 | 95 | 8.941 | 300 | 417.274 |
| 1 | 2030 | 7/23/2005 | 1 | 1 | R-1 | 7.5 | 95 | 9.26 | 300 | 417.593 |
| 1 | 2100 | 10/1/2005 | 1 | 1 | R-1 | 9.5 | 99 | 99.718 | 300 | 534.274 |
| 1 | 2170 | 12/10/2005 | 1 | 1 | R-1 | 9.5 | 99 | 92.505 | 300 | 527.061 |
| 1 | 2240 | 2/18/2006 | 1 | 1 | R-1 | 9.5 | 99 | 91.482 | 300 | 526.038 |
| 1 | 2310 | 4/29/2006 | 1 | 1 | R-1 | 9.5 | 99 | 90.459 | 300 | 525.014 |
| 1 | 2380 | 7/8/2006 | 1 | 1 | R-1 | 9.5 | 99 | 89.436 | 300 | 523.991 |

The easting and northing report the location of the seaward edge of the berm, for use in development of mapped displays with GIS software. Average, Minimum, Maximum, and Standard Deviation of berm width, dune width, and dune height (next section) are displayed, calculated over the interval (e.g., 10 weeks).

| Time | Easting | Northing | Avg Berm | Min Berm | Max Berm | SD Berm | Avg Dune Width | Min Dune Width | Max Dune Width | SD Dune Width |
|---|---|---|---|---|---|---|---|---|---|---|
| 1820 | 1499092 | 463317.2 | 8.158 | 8.015 | 8.302 | 0.097 | 95 | 95 | 95 | 0 |
| 1890 | 1499091 | 463316.9 | 8.478 | 8.334 | 8.622 | 0.097 | 95 | 95 | 95 | 0 |
| 1960 | 1499091 | 463316.6 | 8.797 | 8.653 | 8.941 | 0.097 | 95 | 95 | 95 | 0 |
| 2030 | 1499091 | 463316.3 | 9.116 | 8.973 | 9.26 | 0.097 | 95 | 95 | 95 | 0 |
| 2100 | 1499038 | 463212.3 | 81.842 | 9.292 | 100.128 | 38.229 | 98.2 | 95 | 99 | 1.687 |
| 2170 | 1499041 | 463218.8 | 83.456 | 43.155 | 93.119 | 17.401 | 99 | 99 | 99 | 0 |
| 2240 | 1499042 | 463219.7 | 91.942 | 91.482 | 92.403 | 0.31 | 99 | 99 | 99 | 0 |
| 2310 | 1499042 | 463220.6 | 90.919 | 90.459 | 91.38 | 0.31 | 99 | 99 | 99 | 0 |
| 2380 | 1499043 | 463221.5 | 89.896 | 89.436 | 90.357 | 0.31 | 99 | 99 | 99 | 0 |

| Time | Avg Dune Height | Min Dune Height | Max Dune Height | SD Dune Height | Avg Upland Width | Min Upland Width | Max Upland Width | SD Upland Width | Post Recovery Berm Width Change | Remaining Recovery Berm Width |
|---|---|---|---|---|---|---|---|---|---|---|
| 1820 | 7.5 | 7.5 | 7.5 | 0 | 300 | 300 | 300 | 0 | 0 | 0 |
| 1890 | 7.5 | 7.5 | 7.5 | 0 | 300 | 300 | 300 | 0 | 0 | 0 |
| 1960 | 7.5 | 7.5 | 7.5 | 0 | 300 | 300 | 300 | 0 | 0 | 0 |
| 2030 | 7.5 | 7.5 | 7.5 | 0 | 300 | 300 | 300 | 0 | 0 | 0 |
| 2100 | 9.1 | 7.5 | 9.5 | 0.843 | 300 | 300 | 300 | 0 | 0 | 0 |
| 2170 | 9.5 | 9.5 | 9.5 | 0 | 300 | 300 | 300 | 0 | 0 | 0 |
| 2240 | 9.5 | 9.5 | 9.5 | 0 | 300 | 300 | 300 | 0 | 0 | 0 |
| 2310 | 9.5 | 9.5 | 9.5 | 0 | 300 | 300 | 300 | 0 | 0 | 0 |
| 2380 | 9.5 | 9.5 | 9.5 | 0 | 300 | 300 | 300 | 0 | 0 | 0 |

***Reach2.csv***

This output file reports yearly on the state of the Reach parameters. The following example is filtered on Reach R-1. This file is similar in character to the land loss file (end of year by Reach), with the exception of the

seaward berm retreat, calculated from the initial seaward berm position, and will likely be combined with that file in future releases.

| ItIteration | Time | Date | Reach | Economic Reach | Description | Dune Height | Dune Width | Berm Width | Upland Width | Seaward Berm | Seaward Berm Retreat | Easting | Northing |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1/1/2000 | 1 | 1 | R-1 | 7.5 | 95 | 0 | 300 | 408.333 | 0 | 1499095 | 463324.6 |
| 1 | 366 | 1/1/2001 | 1 | 1 | R-1 | 7.5 | 95 | 1.66 | 300 | 409.994 | -1.66 | 1499095 | 463323.1 |
| 1 | 731 | 1/1/2002 | 1 | 1 | R-1 | 7.5 | 95 | 3.321 | 300 | 411.654 | -3.321 | 1499094 | 463321.6 |
| 1 | 1096 | 1/1/2003 | 1 | 1 | R-1 | 7.5 | 95 | 4.981 | 300 | 413.315 | -4.981 | 1499093 | 463320.1 |
| 1 | 1461 | 1/1/2004 | 1 | 1 | R-1 | 7.5 | 95 | 6.642 | 300 | 414.975 | -6.642 | 1499092 | 463318.6 |
| 1 | 1827 | 1/1/2005 | 1 | 1 | R-1 | 7.5 | 95 | 8.302 | 300 | 416.636 | -8.302 | 1499092 | 463317.2 |

***ReachChangeStatistics.csv***

This output file reports the initial values (start of simulation) and average final values after all iterations, for dune height, dune width, and berm width, and the difference between the initial and final average, by Reach. It shows the overall average morphology change for each Reach (incorporating storm response, planform, and management).

| Reach | Profile | N | DH Existing | DH Average | DHSD | DH Diff | DW Existing | DW Average | DW SD | DW Diff |
|---|---|---|---|---|---|---|---|---|---|---|
| R-1 | R1 | 14335 | 7.5 | 9.1 | 0.9 | 1.6 | 95 | 81.3 | 33.9 | -13.7 |
| R-2 | R2R5 | 14335 | 16.5 | 18.3 | 0.6 | 1.8 | 45 | 53.6 | 4.4 | 8.6 |
| R-3 | R2R5 | 14335 | 16.5 | 18.3 | 0.6 | 1.8 | 45 | 53.6 | 4.3 | 8.6 |
| R-4 | R2R5 | 14335 | 16.5 | 18.3 | 0.6 | 1.8 | 45 | 53.7 | 4.1 | 8.7 |

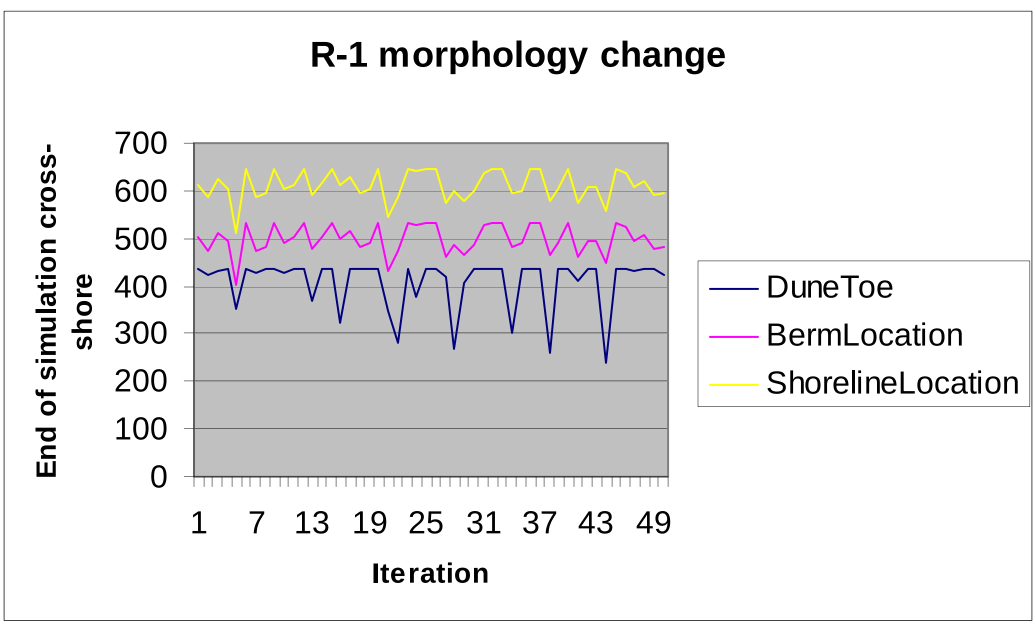
| Reach | BW Existing | BW Average | BW SD | BW Diff |
|---|---|---|---|---|
| R-1 | 0 | 81.2 | 40.2 | 81.2 |
| R-2 | 0 | 52.8 | 29.5 | 52.8 |
| R-3 | 0 | 56.1 | 28.6 | 56.1 |
| R-4 | 0 | 59.6 | 27.8 | 59.6 |

***ReachIteration.csv***

This output file reports end of iteration Reach cross-shore locations for berm, dune toe, and shoreline, and counts of the number of Emergency Nourishments for the Reach in the iteration, for each Reach.

| Iteration | Reach | Number | Dune Toe | Berm Location | Shoreline Location | EN Count | PV EN Cost | Total Volume EN | Total Erosion | Annual Erosion | Structure Damages | Contents Damages |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | R-1 | 1 | 434.556 | 501.371 | 613.616 | 0 | 0 | 0 | 93.03762 | 1.69159 | 2244453 | 4409063 |
| 2 | R-1 | 1 | 422.414 | 473.327 | 585.572 | 1 | 290620 | 9600 | 64.99385 | 1.18171 | 2644784 | 4332145 |
| 3 | R-1 | 1 | 430.536 | 511.292 | 623.537 | 1 | 288631 | 9600 | 102.9587 | 1.87198 | 2501342 | 4401673 |
| 4 | R-1 | 1 | 434.555 | 492.892 | 605.137 | 1 | 340155 | 9600 | 84.55855 | 1.53743 | 10277236 | 8391217 |
| 5 | R-1 | 1 | 351.087 | 400.949 | 513.194 | 1 | 303145 | 9600 | -7.38426 | -0.13426 | 0 | 0 |

An example of a plot that is readily generated from this file, showing variability of simulated shoreline change with iteration, is shown here.



### *ReachStatistics.csv*

The Reach statistics output file contains the same information that is found in the .prn file for end of simulation statistics for annual erosion, structure damage, contents damage, emergency and Planned Nourishments counts, costs, and volumes, by Reach. This simplifies the generation of plots showing Reach variability. For each Reach, four statistics are displayed for each element (average, standard deviation, maximum, and minimum). Note that, at present, the iterations at which the maximum and minimum values occur are not in this file, only in the .prn file.

The following are partial results taken from a 5-iteration run.

| Reach | Annual Erosion Avg | Annual Erosion SD | Annual Erosion Max | Annual Erosion Min | Structure Damage Avg | Structure Damage SD | Structure Damage Max | Structure Damage Min | Contents Damage Avg | Contents Damage SD | Contents Damage Max | Contents Damage Min |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R-1 | 1.23 | 0.804 | 1.872 | -0.134 | 3533563 | 3920472 | 10277236 | 0 | 4306819 | 2968631 | 8391217 | 0 |
| R-2 | 1.474 | 0.342 | 2.048 | 1.228 | 1855.284 | 3334.614 | 7736.41 | 0 | 0 | 0 | 0 | 0 |
| R-3 | 1.61 | 0.315 | 2.13 | 1.38 | 8196.606 | 18328.17 | 40983.03 | 0 | 0 | 0 | 0 | 0 |
| R-4 | 1.757 | 0.284 | 2.214 | 1.531 | 235.696 | 527.031 | 1178.478 | 0 | 0 | 0 | 0 | 0 |

Statistics on the number of Emergency Nourishments, cost, and volume are available (volume not shown in example).

| Reach | Emergency Nourishment Avg | Emergency Nourishment SD | Emergency Nourishment Max | Emergency Nourishment Min | Emergency Nourishment $Avg | Emergency Nourishment $SD | Emergency Nourishment $Max | Emergency Nourishment $Min |
|---|---|---|---|---|---|---|---|---|
| R-1 | 0.8 | 0.447 | 1 | 0 | 244510.3 | 138242.6 | 340155.5 | 0 |
| R-2 | 0.2 | 0.447 | 1 | 0 | 50989 | 114014.9 | 254945 | 0 |
| R-3 | 0.2 | 0.447 | 1 | 0 | 42467.01 | 94959.13 | 212335.1 | 0 |
| R-4 | 0.2 | 0.447 | 1 | 0 | 44352.03 | 99174.16 | 221760.2 | 0 |

Similar information is available in the file for Planned Nourishment (volumes not shown).

| Reach | Planned Nourishment Avg | Planned Nourishment SD | Planned Nourishment Max | Planned Nourishment Min | Planned Nourishment $Avg | Planned Nourishment $SD | Planned Nourishment $Max | Planned Nourishment $Min |
|---|---|---|---|---|---|---|---|---|
| R-1 | 3.4 | 0.548 | 4 | 3 | 2507066 | 445938.6 | 3154445 | 2130759 |
| R-2 | 3.4 | 0.548 | 4 | 3 | 1995031 | 66001.39 | 2060325 | 1905576 |
| R-3 | 3.4 | 0.548 | 4 | 3 | 1600976 | 54207.71 | 1659511 | 1527093 |
| R-4 | 3.4 | 0.548 | 4 | 3 | 1605998 | 56187.58 | 1670855 | 1530298 |

***_ReachYearlyDamages.csv***

This output file shows the yearly total damages in each Reach, and damages assigned to individual loss elements (wave, flooding, erosion) prior to combination, for structures and contents. The example shows present values for structures, but the file itself contains additional information on present values for contents, and raw (non-discounted) damages for structure and contents. Note that this file reports damages every year, whether or not there is damage.

| Iteration | Time | Date | Reach | Economic Reach | Description | Structure Count | Structure Combined PV | Structure Flood Loss PV | Structure Wave Loss PV | Structure Erosion Loss PV |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2922 | 1/1/2008 | 1 | 1 | R-1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 3288 | 1/1/2009 | 1 | 1 | R-1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 3653 | 1/1/2010 | 1 | 1 | R-1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 4018 | 1/1/2011 | 1 | 1 | R-1 | 6 | 1262989 | 205068.3 | 1262989 | 0 |
| 1 | 4383 | 1/1/2012 | 1 | 1 | R-1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 4749 | 1/1/2013 | 1 | 1 | R-1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 5114 | 1/1/2014 | 1 | 1 | R-1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 5479 | 1/1/2015 | 1 | 1 | R-1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 5844 | 1/1/2016 | 1 | 1 | R-1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 6210 | 1/1/2017 | 1 | 1 | R-1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 6575 | 1/1/2018 | 1 | 1 | R-1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 6940 | 1/1/2019 | 1 | 1 | R-1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 7305 | 1/1/2020 | 1 | 1 | R-1 | 5 | 249916.9 | 0 | 249916.9 | 0 |
| 1 | 7671 | 1/1/2021 | 1 | 1 | R-1 | 0 | 0 | 0 | 0 | 0 |

***ReachYearlyDamagesByType.csv***

This output file is similar to the ReachYearlyDamages output previously described, but further broken down by Damage Element Type. Note that, to limit file sizes, information is recorded only in years for which there is in fact damage for elements of the specified type. Similar information is available in the file for contents and non-present valued information, but is not shown in the following example.

| Iteration | Time | Date | Reach | Economic Reach | Description | Type | Structure Count | Structure Combined PV | Structure FloodLoss PV | Structure Wave Loss PV | Structure Erosion Loss PV |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4018 | 1/1/2011 | 1 | 1 | R-1 | MFR1 | 6 | 1262989 | 205068.3 | 1262989 | 0 |
| 1 | 7305 | 1/1/2020 | 1 | 1 | R-1 | MFR1 | 5 | 249916.9 | 0 | 249916.9 | 0 |
| 1 | 12784 | 1/1/2035 | 1 | 1 | R-1 | MFR1 | 6 | 524202 | 154674.8 | 524202 | 0 |
| 1 | 13880 | 1/1/2038 | 1 | 1 | R-1 | MFR1 | 6 | 205170.1 | 4167.903 | 205170.1 | 0 |
| 1 | 13880 | 1/1/2038 | 1 | 1 | R-1 | POOL | 1 | 2175.168 | 0 | 0 | 2175.168 |
| 2 | 3288 | 1/1/2009 | 1 | 1 | R-1 | MFR1 | 6 | 957310.3 | 21959.07 | 957310.3 | 0 |
| 2 | 5844 | 1/1/2016 | 1 | 1 | R-1 | MFR1 | 6 | 1588265 | 427169.1 | 1588265 | 0 |
| 2 | 6940 | 1/1/2019 | 1 | 1 | R-1 | MFR1 | 1 | 99208.38 | 0 | 99208.38 | 0 |
| 3 | 1827 | 1/1/2005 | 1 | 1 | R-1 | MFR1 | 6 | 1339672 | 145760 | 1339672 | 0 |
| 3 | 9862 | 1/1/2027 | 1 | 1 | R-1 | MFR1 | 6 | 942616.2 | 273263.6 | 942616.2 | 0 |
| 3 | 15341 | 1/1/2042 | 1 | 1 | R-1 | MFR1 | 6 | 218109.1 | 29626.64 | 218109.1 | 0 |
| 3 | 15341 | 1/1/2042 | 1 | 1 | R-1 | POOL | 1 | 943.714 | 0 | 0 | 943.714 |
| 4 | 731 | 1/1/2002 | 1 | 1 | R-1 | MFR1 | 6 | 6905356 | 2646657 | 6905356 | 0 |
| 4 | 731 | 1/1/2002 | 1 | 1 | R-1 | POOL | 1 | 6501.821 | 0 | 0 | 6501.821 |
| 4 | 6210 | 1/1/2017 | 1 | 1 | R-1 | MFR1 | 6 | 3365377 | 1247439 | 3365377 | 0 |

***Rebuilding.csv***

This output file reports on each Damage Element when it is rebuilt, giving the initial values, the current values, and the rebuilding cost (difference between current and initial) as well as whether or not the Lot has been condemned. The following example shows selected columns for a single Damage Element. Note that rebuilding is reported when rebuilding is accomplished, but damage is assumed to take place at the time of the storm.

| Iteration | Time | Storm Time | Lot Con-demned | DE TYPE | Initial Structure | Initial Contents | Current Structure | Current Contents | Rebuild Cost Structure | Rebuild Cost Contents | Times Rebuilt |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 4339.772 | 3900.959 | 1 | MFR1 | 1991597 | 995798.5 | 1829143 | 612036 | 162453.8 | 383762.5 | 1 |
| 1 | 7626.541 | 7205.277 | 1 | MFR1 | 1991597 | 995798.5 | 1897207 | 597110.8 | 94390.2 | 398687.7 | 2 |
| 1 | 13144.35 | 12693.07 | 1 | MFR1 | 1991597 | 995798.5 | 1769650 | 565250.5 | 221947.4 | 430548 | 3 |
| 1 | 14171.14 | 13735.82 | 1 | MFR1 | 1991597 | 995798.5 | 1844138 | 622432.7 | 147459.5 | 373365.8 | 4 |

### *StormGeneration.csv*

This output file provides information on each storm that has been used in the simulation, by iteration and start time. Note that this includes both generated storms and selected storms. The "days since previous storm" for the first storm will always be a large number, as there is no valid previous storm for the first storm.

| Iteration | Start Time | Season Number | Storm Type | Date | Storm Identifier | Days Since Previous Storm | Inter Arrival Violation | Month | Day | Day of Year |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2104.949 | 6 | H | 10/5/2005 22:46 | WCe_19641004_H2 | 38628.9 | 0 | 10 | 5 | 278 |
| 1 | 3490.871 | 3 | H | 7/22/2009 20:54 | WCe_18870727_H2 | 1385.9 | 0 | 7 | 22 | 203 |
| 1 | 3550.336 | 5 | H | 9/20/2009 8:03 | WCe_18890923_M3 | 59.5 | 0 | 9 | 20 | 263 |
| 1 | 3576.393 | 6 | H | 10/16/2009 9:25 | WCe_19161018_H3 | 26.1 | 0 | 10 | 16 | 289 |
| 1 | 3900.959 | 5 | H | 9/5/2010 23:00 | WCe_19750923_H1 | 324.6 | 0 | 9 | 5 | 248 |
| 1 | 4569.466 | 3 | H | 7/5/2012 11:10 | WCe_18960707_M1 | 668.5 | 0 | 7 | 5 | 187 |
| 1 | 5354.477 | 4 | H | 8/29/2014 11:26 | WCe_19330802_L2 | 785 | 0 | 8 | 29 | 241 |
| 1 | 5372.392 | 5 | H | 9/16/2014 9:24 | WCe_19530926_L3 | 17.9 | 0 | 9 | 16 | 259 |
| 1 | 5726.115 | 5 | H | 9/5/2015 2:44 | WCe_19850902_H3 | 353.7 | 0 | 9 | 5 | 248 |
| 1 | 6485.847 | 6 | H | 10/3/2017 20:19 | WCe_19851031_M1 | 759.7 | 0 | 10 | 3 | 276 |
| 1 | 7186.515 | 5 | H | 9/4/2019 12:21 | WCe_19600915_M3 | 700.7 | 0 | 9 | 4 | 247 |

### *StormIteration.csv*

This output file reports, for each iteration, statistics for the number of storms by season (average, SD, Max, and Min) and type (0 = extra-tropical, 1 = tropical). The example below is truncated after the 3rd season.

| Iteration | Type | S 1 Average | S 1 SD | S 1 Max | S 1 Min | S 2 Average | S 2 SD | S 2 Max | S 2 Min | S 3 Average | S 3 SD | S 3 Max | S 3 Min |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0364 | 0.1889 | 1 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 | 0 | 0 | 0.0182 | 0.1348 | 1 | 0 | 0.0545 | 0.2993 | 2 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 0 | 0 | 0.0182 | 0.1348 | 1 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 1 | 0 | 0 | 0 | 0 | 0.0545 | 0.2292 | 1 | 0 | 0.0727 | 0.3252 | 2 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0182 | 0.1348 | 1 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 1 | 0 | 0 | 0 | 0 | 0.0545 | 0.2292 | 1 | 0 | 0.1273 | 0.3363 | 1 | 0 |

### *StormResponse.csv*

This output file provides information on storm-associated morphology change from pre-storm to lookup to post-storm to recovery, together with internal values that allow for testing the calculations, and flags indicating whether certain types of changes have taken place. It is in general to be used by the support/development team. A storm response type of 1 is catastrophic. A storm type of 1 is tropical, 0 is extra-tropical. Various fields, such as StormID and StormResponseID, relate to locations in the database, used for debugging.

Storm and Reach information section.

| Iteration | Time | Reach | Storm | Storm Date | Storm Response Type | Storm ID | Storm Type | Storm Response ID |
|---|---|---|---|---|---|---|---|---|
| 1 | 2104.9 | R-1 | WCe_19641004_H2 | 10/5/2005 22:46 | 0 | 314 | 1 | 5327 |
| 1 | 2104.9 | R-2 | WCe_19641004_H2 | 10/5/2005 22:46 | 0 | 314 | 1 | 120787 |
| 1 | 2104.9 | R-3 | WCe_19641004_H2 | 10/5/2005 22:46 | 0 | 314 | 1 | 120787 |
| 1 | 2104.9 | R-4 | WCe_19641004_H2 | 10/5/2005 22:46 | 0 | 314 | 1 | 120787 |
| 1 | 3490.9 | R-1 | WCe_18870727_H2 | 7/22/2009 20:54 | 0 | 14 | 1 | 2387 |
| 1 | 3490.9 | R-2 | WCe_18870727_H2 | 7/22/2009 20:54 | 0 | 14 | 1 | 117279 |
| 1 | 3490.9 | R-3 | WCe_18870727_H2 | 7/22/2009 20:54 | 0 | 14 | 1 | 117279 |
| 1 | 3490.9 | R-4 | WCe_18870727_H2 | 7/22/2009 20:54 | 0 | 14 | 1 | 117279 |

Pre-storm and lookup section.

| Reach | Storm | Pre-Storm Berm Width | Pre-Storm Dune Width | Pre-Storm Dune Height | Pre-Storm Upland Width | Lookup Berm Width | Lookup Dune Width | Lookup Dune Height |
|---|---|---|---|---|---|---|---|---|
| R-1 | WCe_19641004_H2 | 99.718 | 99 | 9.5 | 300 | 100 | 99 | 9.5 |
| R-2 | WCe_19641004_H2 | 99.422 | 55 | 18.5 | 205 | 100 | 55 | 18.5 |
| R-3 | WCe_19641004_H2 | 99.498 | 55 | 18.5 | 205 | 100 | 55 | 18.5 |
| R-4 | WCe_19641004_H2 | 99.522 | 55 | 18.5 | 205 | 100 | 55 | 18.5 |
| R-1 | WCe_18870727_H2 | 73.27 | 99 | 9.5 | 300 | 70 | 99 | 9.5 |
| R-2 | WCe_18870727_H2 | 66.157 | 55 | 18.5 | 205 | 70 | 55 | 18.5 |
| R-3 | WCe_18870727_H2 | 67.755 | 55 | 18.5 | 205 | 70 | 55 | 18.5 |
| R-4 | WCe_18870727_H2 | 69.067 | 55 | 18.5 | 205 | 70 | 55 | 18.5 |

Lookup response and Reach constants.

| Reach | Storm | Lookup Berm Width Change | Lookup Dune Width Change | Lookup Dune Height Change | Lookup Upland Width Change | Upland Elevation | Berm Elevation | Dune Slope | Depth of Closure |
|---|---|---|---|---|---|---|---|---|---|
| R-1 | WCe_19641004_H2 | -61.902 | 0 | 0 | 0 | 6 | 6.6 | 0.18 | 30 |
| R-2 | WCe_19641004_H2 | -68.172 | 0 | 0 | 0 | 6 | 6.6 | 0.106 | 30 |
| R-3 | WCe_19641004_H2 | -68.172 | 0 | 0 | 0 | 6 | 6.6 | 0.106 | 30 |
| R-4 | WCe_19641004_H2 | -68.172 | 0 | 0 | 0 | 6 | 6.6 | 0.106 | 30 |
| R-1 | WCe_18870727_H2 | -36.645 | 0 | 0 | 0 | 6 | 6.6 | 0.18 | 30 |
| R-2 | WCe_18870727_H2 | -40.734 | 0 | 0 | 0 | 6 | 6.6 | 0.106 | 30 |
| R-3 | WCe_18870727_H2 | -40.734 | 0 | 0 | 0 | 6 | 6.6 | 0.106 | 30 |
| R-4 | WCe_18870727_H2 | -40.734 | 0 | 0 | 0 | 6 | 6.6 | 0.106 | 30 |

Additional constants and post-storm results (selected columns).

| Reach | Storm | Maximum Scarping Distance | Reach Berm Width Recovery Factor | Post Storm Berm Width | Post Storm Dune Width | Post Storm Dune Height | Post Storm Upland Width | Post Recovery Berm Width |
|---|---|---|---|---|---|---|---|---|
| R-1 | WCe_19641004_H2 | 27.778 | 90 | 37.817 | 99 | 9.5 | 300 | 93.528 |
| R-2 | WCe_19641004_H2 | 47.17 | 90 | 31.25 | 55 | 18.5 | 205 | 92.604 |
| R-3 | WCe_19641004_H2 | 47.17 | 90 | 31.326 | 55 | 18.5 | 205 | 92.681 |
| R-4 | WCe_19641004_H2 | 47.17 | 90 | 31.351 | 55 | 18.5 | 205 | 92.705 |
| R-1 | WCe_18870727_H2 | 27.778 | 90 | 36.625 | 99 | 9.5 | 300 | 69.605 |
| R-2 | WCe_18870727_H2 | 47.17 | 90 | 25.422 | 55 | 18.5 | 205 | 62.083 |
| R-3 | WCe_18870727_H2 | 47.17 | 90 | 27.021 | 55 | 18.5 | 205 | 63.682 |
| R-4 | WCe_18870727_H2 | 47.17 | 90 | 28.333 | 55 | 18.5 | 205 | 64.994 |

Additional results information.

| Reach | Storm | Shoreline Lookup | Shoreline Lookup Pre Storm | Shoreline Post Storm | Shoreline Shift | Shoreline Shift Pre to Post | Upland Width Change | Peak Surge Plus Tide |
|---|---|---|---|---|---|---|---|---|
| R-1 | WCe_19641004_H2 | 547.8 | 547.519 | 485.617 | 62.183 | -61.902 | N | 2.76 |
| R-2 | WCe_19641004_H2 | 647.434 | 646.855 | 578.683 | 68.75 | -68.172 | N | 2.76 |
| R-3 | WCe_19641004_H2 | 647.434 | 646.931 | 578.76 | 68.674 | -68.172 | N | 2.76 |
| R-4 | WCe_19641004_H2 | 647.434 | 646.956 | 578.784 | 68.649 | -68.172 | N | 2.76 |
| R-1 | WCe_18870727_H2 | 517.8 | 521.07 | 484.425 | 33.375 | -36.645 | N | 3.92 |
| R-2 | WCe_18870727_H2 | 617.434 | 613.59 | 572.856 | 44.578 | -40.734 | N | 3.92 |
| R-3 | WCe_18870727_H2 | 617.434 | 615.189 | 574.455 | 42.979 | -40.734 | N | 3.92 |
| R-4 | WCe_18870727_H2 | 617.434 | 616.501 | 575.767 | 41.667 | -40.734 | N | 3.92 |

Morphology debugging information.

| Reach | Storm | Profile Evolution Calculation Code | Abnormal Response | Case | Type | Condition | Unrestored Catastrophic | Pre-Response Unrestored Catastrophic | Dune Height Forced Below Minimum |
|---|---|---|---|---|---|---|---|---|---|
| R-1 | WCe_19641004_H2 | 13000 | 0 | 13 | LU | 3 | N | N | N |
| R-2 | WCe_19641004_H2 | 13000 | 0 | 13 | LU | 3 | N | N | N |
| R-3 | WCe_19641004_H2 | 13000 | 0 | 13 | LU | 3 | N | N | N |
| R-4 | WCe_19641004_H2 | 13000 | 0 | 13 | LU | 3 | N | N | N |
| R-1 | WCe_18870727_H2 | 13000 | 0 | 13 | LU | 3 | N | N | N |
| R-2 | WCe_18870727_H2 | 13000 | 0 | 13 | LU | 3 | N | N | N |
| R-3 | WCe_18870727_H2 | 13000 | 0 | 13 | LU | 3 | N | N | N |
| R-4 | WCe_18870727_H2 | 13000 | 0 | 13 | LU | 3 | N | N | N |

Additional morphology debugging information.

| Reach | Storm | Dune Rollover | Scarping | Scarp Exceeded | Scarped Amount | Reach Recoverable Berm Width | Dune Overtopping |
|---|---|---|---|---|---|---|---|
| R-1 | WCe_19641004_H2 | N | N | N | 0 | 55.712 | N |
| R-2 | WCe_19641004_H2 | N | N | N | 0 | 61.355 | N |
| R-3 | WCe_19641004_H2 | N | N | N | 0 | 61.355 | N |
| R-4 | WCe_19641004_H2 | N | N | N | 0 | 61.355 | N |
| R-1 | WCe_18870727_H2 | N | N | N | 0 | 32.981 | N |
| R-2 | WCe_18870727_H2 | N | N | N | 0 | 36.661 | N |
| R-3 | WCe_18870727_H2 | N | N | N | 0 | 36.661 | N |
| R-4 | WCe_18870727_H2 | N | N | N | 0 | 36.661 | N |

# Appendix C: Data Definitions

## Armor Types

**Code.** A textual identifier for the Armor Type.

**Description.** Textual description of the Armor Type.

**Number.** Auto-generated or user-defined unique number for the Armor Type, used as an external identifier.

## Configuration Settings

**Tag.** The textual tag associated with the Configuration Setting.

**Description.** A textual description of the purpose for the setting.

**Value.** The value to use for the setting. Note this is the only field editable by the user.

## Construction Types

**Code.** A textual identifier for the Construction Type.

**Description.** Textual description of the Construction Type.

**Number.** Auto-generated or user-defined unique number for the Construction Type, used as an external identifier.

## Damage Elements

**External ID.** An auto-generated or user-defined value that can be used to uniquely identify each Damage Element in the project.

**Description.** A textual description of the Damage Element.

**Type.** Chosen from a drop-down list of all defined Damage Element Types entered into the system (see the "Damage Element Types" section in Chapter 5).

**Foundation Type.** Chosen from a drop-down list of all defined Foundation Types entered into the system (see the "Foundation Types" section in Chapter 5).

**Construction Type.** Chosen from a drop-down list of all defined Construction Types entered into the system (see the "Construction Types" section in Chapter 5).

**Armor Type.** Chosen from a drop-down list of all defined Armor Types entered into the system (see the "Armor Types" section in Chapter 5).

**Length (feet).** Shore perpendicular distance.

**Width (feet).** Shore parallel distance.

**Number of Floors.** Number representing the total number of floors in the Damage Element.

**Time to Rebuild (rebuild attributes).** Triangular distribution of rebuild time in days (minimum, most likely, maximum).

**Number of Rebuilds (rebuild attributes).** Maximum number of times the Damage Element can be rebuilt.

**Representative Point Easting.** The X coordinate of the Damage Element.

**Representative Point Northing.** The Y coordinate of the Damage Element.

**Contents Value.** Triangular distribution of contents value (minimum, most likely, maximum).

**Structure Value.** Triangular distribution of structure value (minimum, most likely, maximum).

**First Floor Elevation.** Triangular distribution of First Floor Elevation (minimum, most likely, maximum).

## Damage Element Types

**Code.** A textual identifier for the Damage Element Type.

**Description.** Textual description of the Damage Element.

**Number.** Auto-generated or user-defined unique number for the Damage Element Type, used as an external identifier.

**Linear Element.** A Boolean flag indicating whether the element is considered to be linear such as pools, walkways, tennis courts, etc. Damage Elements defined as linear elements should also be marked as auto locate elevation Damage Elements.

**Linear Element Division Length (feet).** Linear elements can be divided into sections. This parameter specifies to length of each section for each specific Damage Element Type.

**Auto Locate Elevation.** A Boolean flag indicating whether a Damage Element of a specific type should have its elevation calculated automatically based on the elevation of the parent Profile that particular location.

**Elevation Offset (feet).** An integer value representing an offset distance to be applied to a Damage Element's elevation if that Damage Element is of that specific type. Used for linear and auto locate elevation Damage Elements.

**Elevation Variability (feet).** This input expresses the uncertainty in the Damage Element elevation attribute specification and is used to establish the minimum and maximum elevation specifications for Damage Element Types marked as linear and auto locate elevation Damage Elements.

**Condemnation Ratio.** The ratio of post-storm structure value divided by initial structure value below which will result in the structure being marked as condemned, provided that the Damage Element Type is also marked as condemnable. For example, if a condemnation ratio of 0.7 is specified for Single-Family Residential (SFR1) structures and SFR1 Damage Element Types are marked as "condemnable" structures then, a SFR1 Damage Element will be marked as condemned if a storm produces damages that exceed 30 percent of the of the initially specified structure value.

**Condemnable Flag.** A Boolean flag indicating whether Damage Elements of that type are condemnable.

## Damage Functions

**Type (read-only).** The specific type of damage function. The six different types of functions are predetermined and the listed type depends on the type node that is selected in the tree.

**Name.** The customized name for the function. When a new function is added, this will be used as the display name of the generated node in the object tree. Should be an abbreviated short name for the function.

**Description.** A more detailed textual description of what the function represents.

**X-axis.** User entered numeric values for % Footprint Compromised (Erosion), or Water Depth above First Floor (Inundation and Wave).

**Y-axis (minimum, most likely, maximum).** User entered numeric values for the minimum, most likely, and maximum factional damage to contents/structure.

NOTE: All functions must be closed, meaning each Y value (minimum, most likely, maximum) must be 0 for some X value.

## Foundation Types

**Code.** A textual identifier for the Foundation Type.

**Description.** Textual description of the Foundation Type.

**Number.** Unique number for the Foundation Type, used as an external identifier.

**Critical Erosion Amount (feet).** Vertical erosion that will compromise this Foundation Type.

**Shallow Foundation.** A Boolean flag that indicates whether or not foundations of that type are shallow foundations.

## Emergency Nourishment Alternatives

**Name.** A textual identifier for the Emergency Nourishment Alternative.

**Description.** Textual description.

**Mobilization Cost ($).** Project-level Mobilization Cost, per nourishment action.

## Lots

**External ID.** An auto-generated or user-defined value that can be used to uniquely identify each Lot in the project.

**Description.** A textual or numeric description of the Lot.

**Type.** A drop-down list containing all valid Lot Types. Currently a Lot can be either Residential or Vacant.

**Armoring Status.** Selected from a drop-down list containing all valid choices: Armorable In The Future, Already Armored, or Not Armorable.

**Erosion (armor failure threshold).** Magnitude (feet) of vertical erosion at the cross-shore location of the armor unit that will cause the armor to fail.

**Flooding (armor failure threshold).** Water-surface elevation at the cross-shore location of the armor unit that will cause the armor to fail.

**Wave Damage (armor failure threshold).** Wave height at the cross-shore location of the armor unit that will cause the armor to fail.

**Distance Trigger (armor construction).** Offset distance (feet) between the seaward edge of the berm and the seaward-most Lot corner that will trigger armor construction on the Lot.

**Length (armor construction).** Length of armor to be constructed on Lot (feet).

**Mobilization Cost (armor construction).** All costs associated with armor construction not included in the Armor Construction Cost Per Foot

specification (e.g., engineering and design, equipment rental, backfill material, etc.).

**Cost Per Foot (armor construction).** Estimated cost of armor construction per foot of armor length.

**Mobilization Time (armor construction).** Estimate of time lag between trigger for armor construction and actual initiation of armor construction.

**Time Per Foot (armor construction).** Estimate of the time required for constructing armor, expressed as days per foot of armor length.

**P1 Northing, P1 Easting.** The Northing and Easting coordinates for point P1 (Figure 28).

**P2 Northing, P2 Easting.** The Northing and Easting coordinates for point P2 (Figure 28).

**P3 Northing, P3 Easting.** The Northing and Easting coordinates for point P3 (Figure 28).

**P4 Northing, P4 Easting.** The Northing and Easting coordinates for point P4 (Figure 28).

## Nourishment Blackout Windows

**Description.** Textual description for the blackout window.

**Start Month.** Month the blackout window starts.

**Start Day.** Day the blackout window starts.

**End Month.** Month the blackout window ends.

**End Day.** Day the blackout window ends.

## Output Options

**Output Flag.** A Boolean flag indicating whether a specific output file should be generated by the Kernel when the simulation is run.

**Sort Order.** User specified integer value, to change placement of rows in the table.

**Information Fields.** Not modifiable by user.

**Tag.** Not modifiable by user. A condensed string tag associated with each specific output data file.

**Description.** Not modifiable by user. A detailed textual description of what data the output file will contain.

**Generation.** Not modifiable by user. A brief description of when the associated output is generated during the simulation.

**File Suffix.** Not modifiable by user. Files generated by the Kernel have the naming convention of <ScenarioName><FileSuffix>. The file suffix includes the portion of the name that uniquely identifies it as well as the associated file extension type.

## Planned Nourishment Alternatives

**Name.** A textual identifier for the Planned Nourishment Alternative.

**Description.** A textual description which typically provides more detail than the name.

**Start Date.** The date the Nourishment Alternative goes into affect.

**Time Increment (years).** Planned renourishment cycle (i.e. the time between Planned Nourishment events).

**Mobilization Cost ($).** Mobilization Cost, per nourishment.

**Default Borrow To Placement Ratio.** Estimated volume of borrow material required to produce a unit volume of stable fill material on the project beach, Reach-specific borrow to placement ratio set at the Reach level.

**Mobilization Threshold.** Minimum number of cubic yards of nourishment material to be placed to justify project mobilization costs.

**Type.** Periodic-Tested. This nourishment implementation type assumes testing on a regular cycle. If at the time of testing the volumetric nourishment need is less than the mobilization threshold, then testing for project mobilization is not reattempted until the next Planned Nourishment cycle. Periodic-Tested is currently the only nourishment type implemented in Beach-*fx*. Additional nourishment implementation types are planned to be added in future versions.

## Profiles

**Number.** An auto-generated or user-defined integer value greater than zero that can be used to uniquely identify each Profile in the project.

**Description.** User-defined description of the Profile that must match the Profile name used in the SBEACH simulations.

**Default Dune Height (feet).** The height of the dune, measured from datum (elevation 0).

**Default Dune Width (feet).** The width of the top of the dune in feet.

**Default Berm Height (feet).** The height of the berm, measured from datum (elevation 0).

**Default Berm Width (feet).** The width of the top of the berm.

**Dune Slope.** The side slope of the dune. The landward and seaward dune slope is assumed to be constant and equal.

**Foreshore Slope.** The beach slope from the berm to datum.

**Upland Elevation (feet).** Representative ground elevation landward of the dune, measured from datum (elevation 0).

**Depth Of Closure (feet).** Depth below datum to which nourishment material is distributed.

**Detailed Submerged Profile.** A Boolean flag indicating whether the Profile's submerged profile is defined by a set of user-specified X, Z data points or if a representative equilibrium profile is generated instead.

**Submerged Profile Parameter.** A user-specified sediment scale parameter used in calculating the equilibrium profile if the Detailed Submerged Profile Flag is not set. The equilibrium profile is defined by the relationship $Depth = Ax^{2/3}$ where A is the empirical sediment scale parameter and $x$ is the distance offshore from the shoreline.

## Reaches

**Number.** An auto-generated or user-defined integer value greater than zero that can be used to uniquely identify each Reach in the project.

**Description.** A user-defined name for the Reach.

**Length (feet).** The shore-parallel distance represented by the Reach.

**Applied Erosion Rate.** Feet/year erosion (-) or accretion (+), calibration parameter. The expected rate of shoreline change in the absence of storm events.

**Back-Bay Flooding.** A Boolean flag indicating whether or not back-bay flooding is possible for the Reach.

**Planned Nourishment.** A Boolean flag indicating whether or not the Reach will receive Planned Nourishment.

**Emergency Nourishment.** A Boolean flag indicating whether or not the Reach will receive Emergency Nourishment.

**Upland Width (feet).** The width of the upland area behind the dune.

**Flooding Threshold.** The threshold elevation at which back-bay flooding is initiated.

**Economic Reach Number.** The economic Reach number, for output purposes only.

**Control Line Offset.** Threshold distance in feet measured from Lot centroid to the seaward toe of the dune at which Lots in the Reach will be marked as condemned prohibiting the rebuilding of Damage Elements in that Lot.

**Start Point Easting.** GIS coordinate of X (easting) of Reach Start Point on Reference Line (Point 1, Figure 27).

**Start Point Northing.** GIS coordinate of Y (northing) of Reach Start Point on Reference Line (Point 1, Figure 27).

**End Point Easting.** GIS coordinate of X (easting) of Reach End Point on Reference Line (Point 2, Figure 27).

**End Point Northing.** GIS coordinate of Y (northing) of Reach End Point on Reference Line (Point 2, Figure 27).

**Shoreward Start Point Easting.** GIS coordinate of X (easting) of Reach Start Point Shoreward (Point 3, Figure 27).

**Shoreward Start Point Northing.** GIS coordinate of Y (northing) of Reach Start Point Shoreward (Point 3, Figure 27).

**Shoreward End Point Easting.** GIS coordinate of X (easting) of Reach End Point Shoreward (Point 4, Figure 27).

**Shoreward End Point Northing.** GIS coordinate of Y (easting) of Reach End Point Shoreward (Point 4, Figure 27).

**SBEACH Landward Boundary Easting.** GIS coordinate of X (easting) of SBEACH Landward Boundary.

**SBEACH Landward Boundary Northing.** GIS coordinate of Y (northing) of SBEACH Landward Boundary.

**SBEACH Seaward Boundary Easting.** GIS coordinate of X (easting) of SBEACH Seaward Boundary.

**SBEACH Seaward Boundary Northing.** GIS coordinate of Y (northing) of SBEACH Seaward Boundary.

**Berm Width Recovery Factor.** Percent of storm-induced berm width change that is restored due to post-storm recovery processes.

## Reach level Emergency Nourishment

**Emergency Nourishment Alternative.** The Emergency Nourishment Alternative to be assigned to the parent Reach. This is selected from a drop-down list containing all Emergency Nourishment Alternatives defined at the project level.

**Unit Placement Cost ($).** The estimated cost of constructing an Emergency Nourishment project expressed as a cost per cubic yard of fill material.

**Borrow To Placement Ratio.** The estimated volume of borrow material required to produce a unit volume of stable fill material on the project beach. This ratio, often referred to as the overfill ratio and accounts for volumetric losses due the sorting and winnowing of fines contained in the fill material.

**Production Rate.** The rate at which fill volume is placed on the beach to construct the Emergency Nourishment project expressed in units of cubic yards per day.

**Mobilization Cost ($).** The Reach-specific costs associated with the nourishment event not included in the Unit Placement Cost attribute.

**Mobilization Time (days).** The estimated time lag between the triggering event and the initiation of Emergency Nourishment construction.

**Priority Order.** An integer value which will determine which Reach receives nourishment first, if more than one Reach is triggered for Emergency Nourishment at the same time. Ordering should begin at 1 and assigned values should be unique.

**Dune Height (feet) (Emergency Nourishment Trigger).** Currently an inactive attribute.

**Dune Width (feet) (Emergency Nourishment Trigger).** A specified dune width that will trigger the first Emergency Nourishment. Subsequent Emergency Nourishments are triggered when the post Emergency Nourishment dune width is further reduced by an amount exceeding the "Emergency Nourishment Trigger Adjustment" value specified in the Configuration Settings table (see the "Emergency Nourishment configuration settings" in Chapter 6).

**Berm Width (feet) (Emergency Nourishment Trigger).** Currently an inactive attribute.

**Dune Height (feet) (Emergency Nourishment Template).** Currently an inactive attribute.

**Dune Width (feet) (Emergency Nourishment Template).** Currently an inactive attribute.

**Berm Width (feet) (Emergency Nourishment Template).** Currently an inactive attribute.

## Reach level Planned Nourishment

**Planned Nourishment Alternative.** The Planned Nourishment Alternative to be assigned to the Reach. This is selected from a list of Planned Nourishment Alternatives the user defined at the project level.

**Unit Placement Cost ($).** The estimated cost of constructing a Planned Nourishment project expressed as a cost per cubic yard of fill material.

**Borrow To Placement Ratio.** The estimated volume of borrow material required to produce a unit volume of stable fill material on the project beach. This ratio, often referred to as the overfill ratio and accounts for volumetric losses due the sorting and winnowing of fines contained in the fill material.

**Production Rate (cubic yards/day).** The rate at which fill volume is placed on the beach to construct the Planned Nourishment project expressed in units of cubic yards per day.

**Processing Order.** An integer value indicating the order Reaches will be processed for nourishment if multiple Reaches are set to receive nourishment at the same time.

**Mobilization Cost ($).** The costs associated with Reach-specific mobilization costs related to the nourishment event and not included in the Unit Placement Cost attribute.

**Dune Height (feet) (nourishment trigger).** Fractional amount of template dune height that denotes requirement for renourishment.

**Dune Width (feet) (nourishment trigger).** Fractional amount of template dune width that denotes requirement for renourishment.

**Berm Width (feet) (nourishment trigger).** Fractional amount of template berm width that denotes requirement for renourishment.

**Dune Height (feet) (template).** The post-construction dune height.

**Dune Width (feet) (template).** The post-construction dune width.

**Berm Width (feet) (template).** The post-construction berm width.

## Reach Planform Rate

**Planned Nourishment Alternative.** The Planned Nourishment Alternative specified for the Reach. This is selected from a list of Planned Nourishment Alternatives the user defined at the project level.

**Nourishment Cycle.** An integer representing the nourishment cycle. Because beach nourishment results in the construction of a sacrificial structure that evolves over time beach nourishment projects are typically composed of multiple beach nourishment construction events. Each construction event represents the beginning of a new nourishment cycle.

**Berm Width Change Rate.** The rate of project-induced shoreline change (feet/year) resulting from the placement of beach nourishment material. Estimates of this input are typically obtained from shoreline change model results. This input is expected to vary spatially by Reach and over time by nourishment cycle.

## Scenarios

**Name.** Scenario names should be unique.

**Description.** A textual description of the Scenario.

**Start Year.** The simulation start year for the Scenario.

**Start Month.** The simulation start month of the Scenario.

**Base Year.** Reference year for present value calculations. This must be greater than the simulation Start Year.

**Emergency Nourishment Flag.** A Boolean flag indicating whether Emergency Nourishment will be applied for the Scenario during the simulation.

**Planned Nourishment Alternative.** Defined Planned Nourishment Alternative that will be applied for this Scenario simulation.

**Emergency Nourishment Alternative.** Defined Emergency Nourishment Alternative.

**Step Flag.** A Boolean flag indicating if the simulation should enter Step Mode to allow the user to single step the simulation.

**Iterations.** The number of life-cycles to be performed.

**Duration.** The number of years spanned in each iteration / life cycle. The life cycle duration should equal the number of years between the simulation start year and the base year (the year the Planned Nourishment Alternative is in place and producing benefits) plus the economic analysis horizon (typically 50 years for the U. S. Army Corps of Engineers Hurricane and Storm Damage Reduction projects).

**Interest.** The interest rate for present value calculations.

**Seed.** Integer seed value for the random number generator which is used to generate the storm sequence during a simulation. It is recommended that the seed be specified as a large prime number. It is important to understand that the declared seed value for successive simulations must be identical in order for those simulations to experience the same sequence of storm events. If different Scenarios are to be inter-compared (e.g., with and without project) the seed value must be the same to ensure the same random sequence of storms is encountered in each Scenario.

**Scarping.** Include detailed treatment of dune scarping (recommended).

**Calibration.** If checked, simulation will involve only morphology change calculations and preclude all economic related calculations.

## Specific Storms

**Storm.** The unique storm identifier. This is selected from the list of existing storms.

**Date Of Storm.** The exact date the storm should occur during a simulation.

**Active.** A Boolean flag indicating whether or not the specific storm should be counted in the pool of active storms. Only active storms will be considered.

## Storms

**Identifier.** A textual name/description that uniquely identifies the storm. This must be identical to the storm name used in the SBEACH simulations.

**Type.** The type of storm (Tropical/Extra-Tropical).

**Relative Probability.** The relative probability between the storms in the plausible storm suite. For instance a storm with a relative probability of 2 is twice as likely to be selected as a storm with a relative probability of 1.

**Peak Surge Plus Tide.** The elevation of flooding from the back bay.

**Date of Storm.** The historical date of the storm. This date is used to assign the storm to a defined Storm Season.

## Storm Seasons

**Number.** Unique identifier.

**Description.** Textual description for the season (i.e., Extra Tropical Only).

**Start Month.** Month the season starts.

**Start Day.** Day the season starts.

**End Month.** Month the season ends.

**End Day.** Day the season ends.

**Previous Season Overlap (days).** The number of days prior to the specified season start month and day for purposes of identifying the population of storm from which the random sample will be taken.

**Next Season Overlap.** The number of days after the specified season end month and day for purposes of identifying the population of storms from which the random sample will be taken.

**Probability Of Extra Tropical Storm.** Probability of occurrence of Extra-Tropical storms during that season.

**Probability Of Tropical Storm.** Probability of occurrence of Tropical storms during that season.

**Minimum Storm Arrival Time (days).** Minimum inter-arrival time for storms in season.

**Probability Active.** Boolean flag activating or deactivating storms in season.

**Maximum Extra-Tropical Storms In Season.** The maximum number of extra-tropical storms that can occur in the season.

**Maximum Tropical Storms In Season.** The maximum number of tropical storms that can occur in the season.

# Appendix D: Training Exercises


# Training Exercises

**October 2007**

# Introduction to Beach-*fx*

The U.S. Army Engineer Research and Development Center (ERDC) in coordination with the Institute for Water Resources (IWR) has developed Beach-*fx* as a planning tool for the analysis of the physical performance and economic benefits and costs of shore protection projects, in particular beach nourishment along sandy shores. Beach-*fx* links the predictive capability of coastal process simulation models with project area infrastructure information (structure inventory), structure and content damage functions, and economic valuation to estimate the costs and benefits of alternative project designs. Beach-*fx* has been implemented as an event based Monte Carlo life cycle simulation model which enables quantification of risk and uncertainty in the resulting estimates of project performance, nourishment volume requirements, as well as expected project costs and estimated with- and without-project damages.

Beach-*fx* is complex in that it draws together theory and data from numerous specialist fields and disciplines into a single modeling system. For example, the physical evolution of the simulated project is estimated based on specific beach morphology responses (dune height change, berm width change, etc.) obtained from a pre-computed database populated from SBEACH simulations. Setting up and executing the required SBEACH simulations is without question an engineering or coastal specialist function, whereas development and attribution of the structure inventory and damage functions is more appropriately a function for an economist or planner. Although responsibilities for certain tasks appear well defined by their very nature, within Beach-*fx* the lines become somewhat blurred and ideally engineers, economists, and planners and other disciplines as necessary, will work together in a collaborative environment to set-up, calibrate and run Beach-*fx* as well as evaluate and interpret the generated output.

The exercises and examples outlined in this document are intended to familiarize the user with the Beach-*fx* interface and input data requirements. The user will be guided through the process of creating a new project, importing pre-compiled input data sets, defining damage function—Damage Element relationships, populating the Storm Response Database (SRD), creating simulation Scenarios, calibrating Beach-*fx*, defining alternative emergency and Planned Nourishment schemes, and executing with and without project simulations.

# 1.1 Beach-*fx* Data Elements

The data elements that comprise a Beach-*fx* project can be categorized into one of five basic types:

- Meterorologic data. (The environmental force that drives shoreline and beach profile evolution.)
  - Plausible storm suite
  - Storm Seasons

- Coastal morphology data. (Representation of the beach morphology and its response to storm-induced and long-term processes.)
  - Profiles
  - Reaches
  - Shore Response Database (SRD)

- Economic data. (Representation of the structure inventory, its attributes and damage relationships.)
  - Lots
  - Damage Elements
  - Damage Functions

- Management measures. (Representation of with- and without-project management approaches to minimize damages due to storm-induced and long-term coastal erosion.)
  - Emergency Nourishment
  - Planned/Scheduled Nourishment

- Model Configuration and Output Specifications.

Beach-*fx* establishes linkages between and among these data types, which entails substantial information, to provide a generalized framework in which the characteristics of the project study area are defined in terms of beach morphology and its response to coastal storms and long-term processes, the level and density of upland development and its economic value and vulnerability to damage, as well as alternative management approaches for protecting the infrastructure and combating the effects of coastal erosion.

## 1.2 Overview of Example Project

The training exercises illustrated here are designed to provide new Beach-*fx* users with a basic understanding of the Beach-*fx* interface, familiarize the user with the input data requirements and outline the procedures and steps required to define a project within Beach-*fx*. The exercises are centered about the example project illustrated in Figure 1.

It is assumed that the coastal processes analysis for the example project area has been completed and the structure inventory has been digitized and attributed within a GIS system using the available Beach-*fx* ESRI shapefile templates.


Figure 1. Example project area.

The following paragraphs summarize the products of the coastal processes analysis as well as the structure inventory and planned lay-out of the project within Beach-*fx*.

### 1.2.1 Coastal Processes Analysis

Through an analysis of the available historical beach profile data, the beach morphology within the project area is characterized by two representative beach Profiles. The northwestern third of the project is characterized by a wide and very low dune feature fronting the condominium complex know as Pinnacle Port, whereas the residential community encompassing the southeastern two thirds of the project area is fronted by a comparatively high dune with mild side slopes. The above datum portions of the two representative Profiles are illustrated in Figure 2.

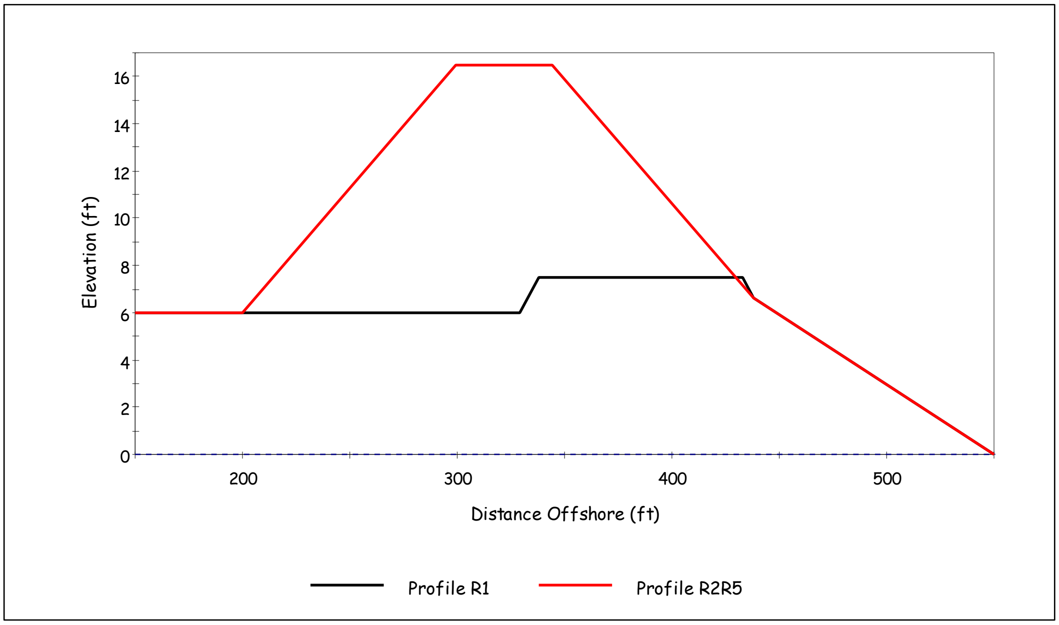



Figure 2. Representative Profiles.

The project area is located in the northern Gulf of Mexico and is impacted by relatively frequent tropical storm events. An analysis of the historical storm climatology resulted in identification of 48 historical storms that generated at least 1 ft of surge between 1886 and 2005 giving an annual probability of tropical storm occurrence of 0.4. These 48 historical storms were expanded to a plausible storm suite consisting of 576 storms by combining the historical storm surge hydrograph with three statistically defined tidal ranges (high, mean, and low) and combining the storm surge hydrograph at four phases of the astronomical tide such that peak surge is aligned at high tide, mid-tide falling, low tide, and mid-tide rising. In terms of relative probability of occurrence, those plausible storms

associated with mean tidal ranges are given a relative probability of 2 whereas those storms associated with high and low tidal ranges are given a relative probability of 1.

SBEACH simulations involving 126 variations of representative Profile R1 and 108 variations of representative Profile R2R5 and the plausible storm suite were performed for a total of 134,784 unique beach profile/storm simulations. The beach profile responses estimated in these SBEACH simulations are used to populate the shore response database (SRD) that becomes input to Beach-*fx*. Within Beach-*fx*, the SRD serves as a look-up table for beach profile response to randomly selected storms from the plausible storm suite. Thus, the full range of anticipated profile configurations that are expected to exist under different with- and without-project conditions should be reasonably covered by the SBEACH simulations. For these exercises, we will assume that this requirement has been met.

### 1.2.2 Project Lay-Out

A project within Beach-*fx* is defined according to a geographical hierarchy consisting of Reaches, Lots and Damage Elements (structure inventory) as illustrated schematically in Figure 3. Development of the required structure inventory, Lot and Reach specifications for Beach-*fx* is believed to be a task well suited for accomplishment within a GIS framework. As such, the Beach-*fx* Model developers have developed three ESRI shapefile templates for compiling and attributing these input data for subsequent import into Beach-*fx*. Template shapefiles are provided in the Beach-*fx* installation package for Damage Elements, Lots, and Reaches.

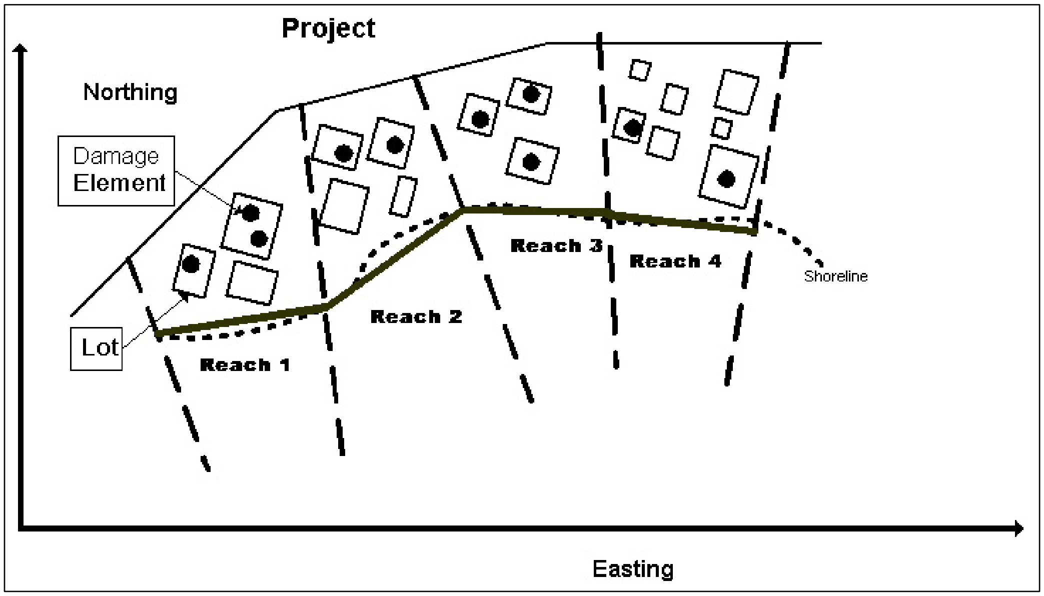



Figure 3. Beach-*fx* schematization of the project study area.

**Damage elements**. Figure 4 is a screen capture of the 273 Damage Elements that comprise the structure inventory identified for the example project.

Figure 4. Damage Elements.

Damage Elements are rectangles represented as point objects with length and width attributes within the shapefile template, “DamageElements.shp” and the coordinates of each point object should represent the approximate center of the structure being identified. Beach-*fx* requires attribute information for each Damage Element (construction material, Foundation Type, first floor elevation, depreciated replacement value, etc.) and this information is provided by the user by filling out the pre-defined attribute table provided in the shapefile template. A complete listing of the required Damage Element attributes and their definitions is provided in the Beach-*fx* User’s Manual. Figure 5 provides a screen capture of the Damage Element attribute table.

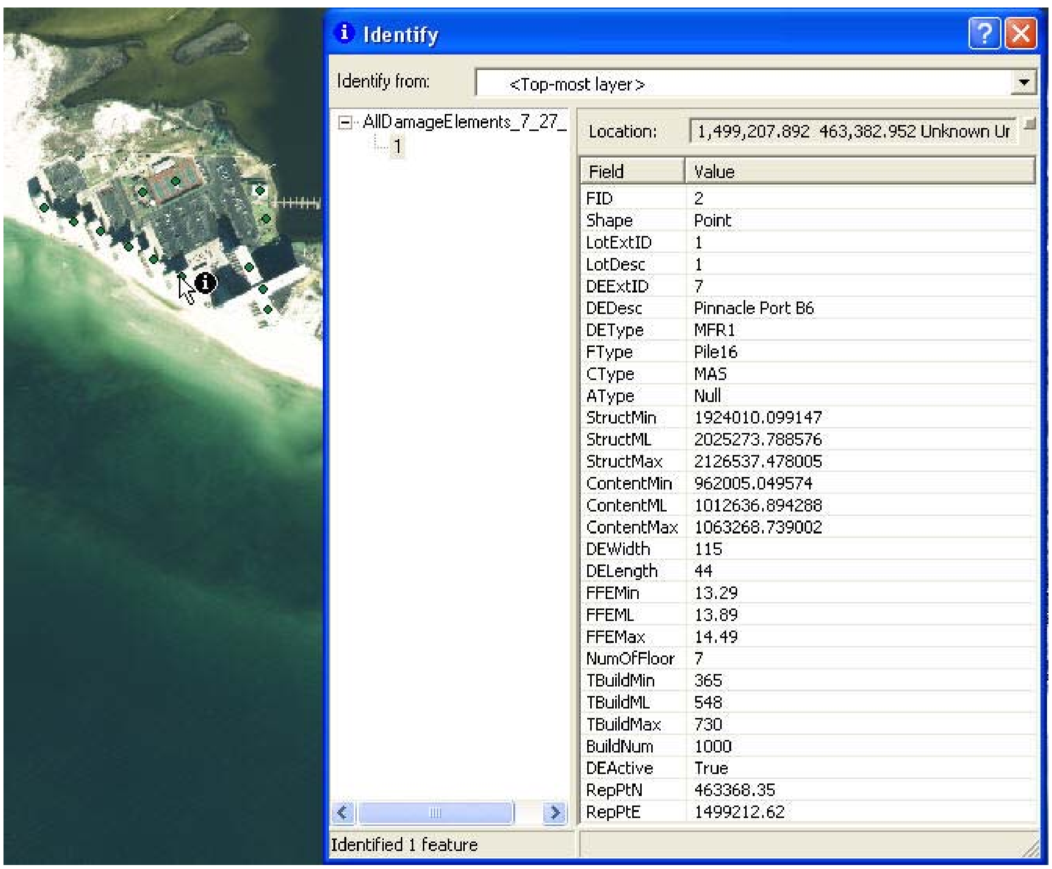



Figure 5. Damage element attribute table.

**Lots.** Within Beach-*fx*, each Damage Element must reside within a defined Lot. Within the GIS, Lots are quadrilateral polygon objects (shapefile template “Lots.shp”) and the example project involves a total of 96 Lots as illustrated in Figure 6. Within the context of Beach-*fx* Lots are containers for Damage Elements. Each Lot is also associated with a series of

attributes as defined in the Beach-*fx* User's Manual. Figure 7 provides a screen capture of the Lot attribute table.

Figure 6. Example project Lots.

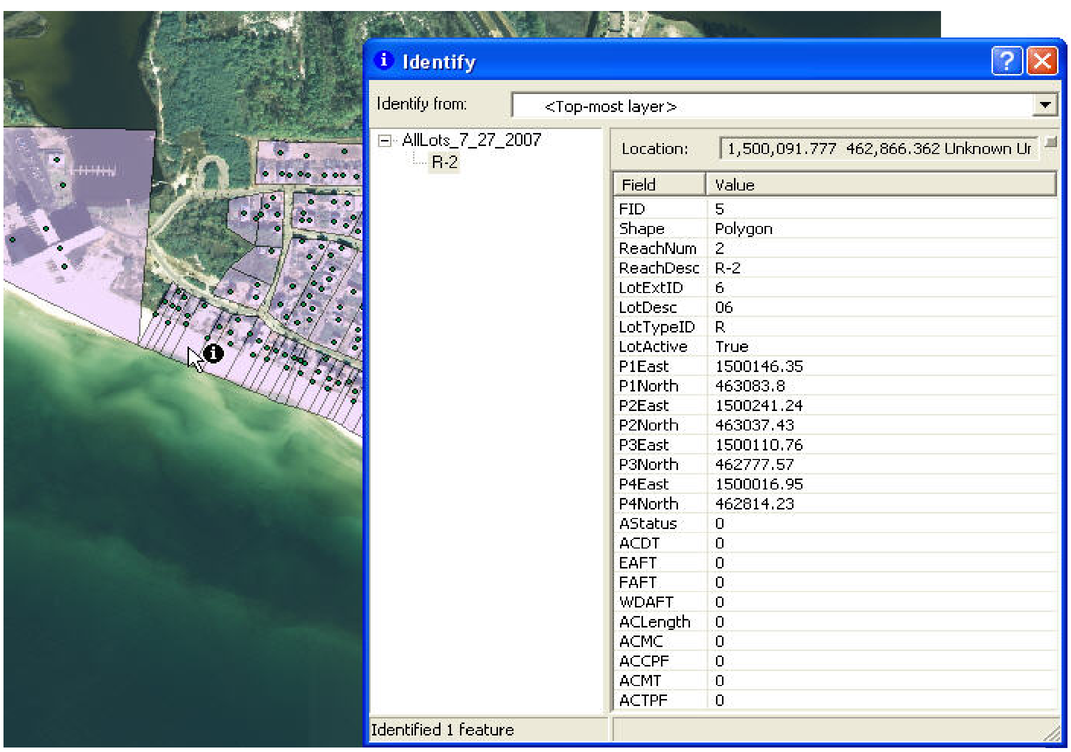



Figure 7. Lot attribute table.

**Reaches.** Within the Beach-*fx* geographical hierarchy Reaches are contiguous, morphologically homogeneous areas. That is, a single representative beach Profile characterizes the entire beach within a Reach. As Lots are containers for Damage Elements, so Reaches are containers for Lots. Each Lot must reside within a defined Beach-*fx* Reach. The shoreline within a Reach is assumed to be straight and perpendicular to the defined SBEACH reference line start and end points. Within the GIS,Reaches are polygon objects (shapefile template "Reaches.shp") and the four Reaches defined for the example project are illustrated in Figure 8. A screen capture of the Reach attribute table is shown in Figure 9. The Reach attributes are defined in the Beach-*fx* User's Manual.

Figure 8. Example project Reaches.

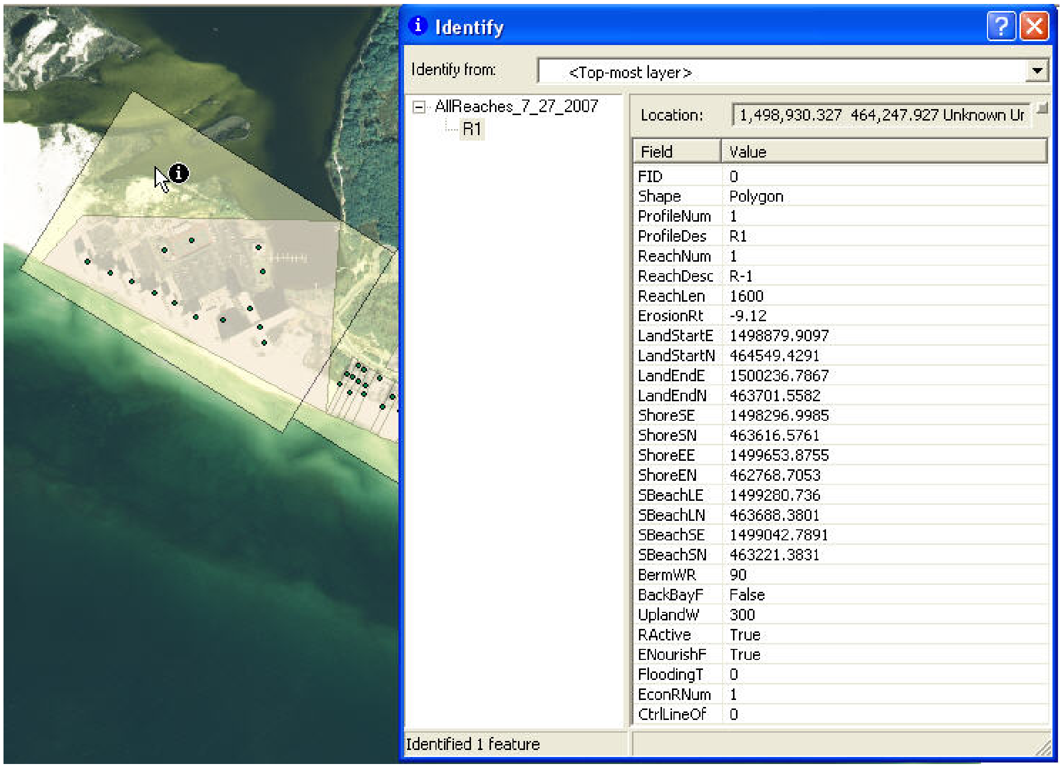



Figure 9. Reach attribute table.

## 1.3 Overview of training exercises

Through the following sequence of training exercises the previously described project study area is created as a Beach-*fx* project. After population of the project data sets, the procedure for calibrating Beach-*fx* is exercised. Additional exercises in which management alternatives including Emergency Nourishment (without-project conditions) and Planned Nourishment (with-project conditions) will be defined and simulations performed.

The hands-on exercises are presented in the following sequence:

Part I: Create and populate new project.

Exercise 1: Create a new blank project.

Exercise 2: Import storms and define Storm Seasons.

Exercise 3: Define initial condition representative Profiles and import Detailed Submerged Profiles.

Exercise 4: Define Reaches and Lots.

Exercise 5: Check/verify Damage Element Variables and define Damage Element inventory.

Exercise 6: Create and populate GIS folder.

Exercise 7: Define Damage Functions.

Exercise 8: Populate Damage Function matrix.

Exercise 9: Build/populate Shore Response Database (SRD).

Part II: Beach-*fx* Calibration.

Exercise 10: Confirm role of applied erosion rate (AER).

Exercise 11: Identify storm-induced rate of shoreline change.

Exercise 12: Perform calibration simulations, iterative adjustment of AER to cause Beach-*fx* to return target historical erosion rate over multiple lifecycle simulations.

Part III: Without-Project Simulations.

Exercise 13: Define Emergency Nourishment Alternative.

Exercise 14: Define without-project Scenario parameters, Output Options and Configuration Settings. Perform without-project simulation.

Part IV: With-Project Simulations.

Exercise 15: Define Planned Nourishment Alternative(s).

Exercise 16: Define with-project Scenario parameters, Output Options and Configuration Settings. Perform with-project simulation.

# Training Exercises

# Part I

# Create and Populate Example Project

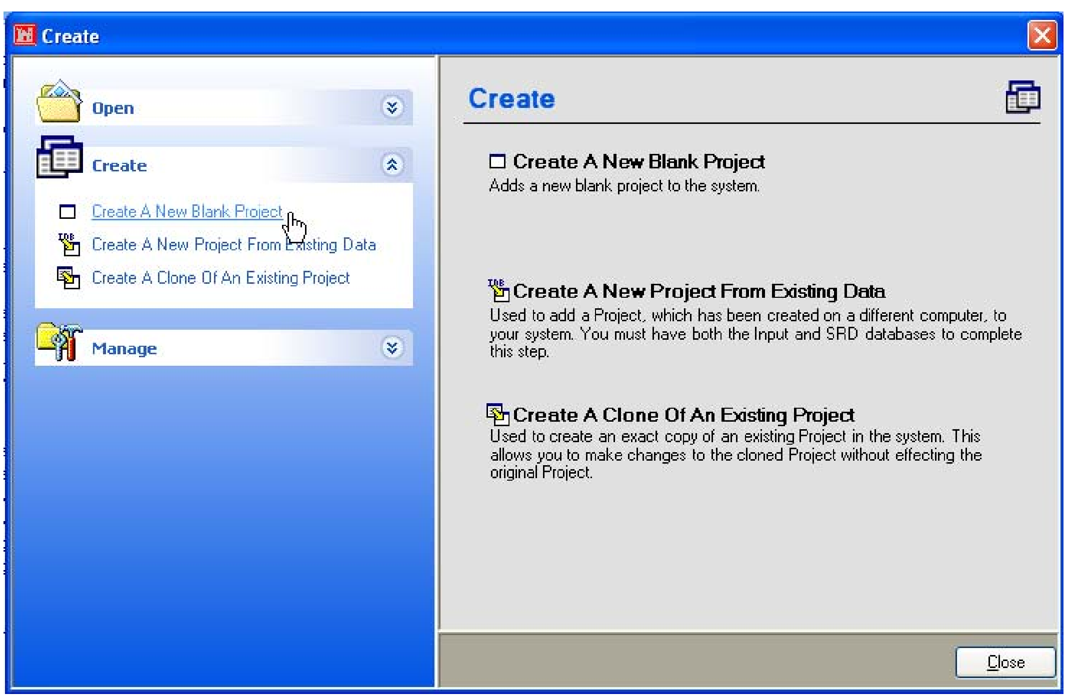
# Exercise 1
# Create a New Blank Project

This exercise illustrates how to initiate development of a new Beach-*fx* Project.

## 1.1 Use Beach-*fx* Project Manager to Create a New Project

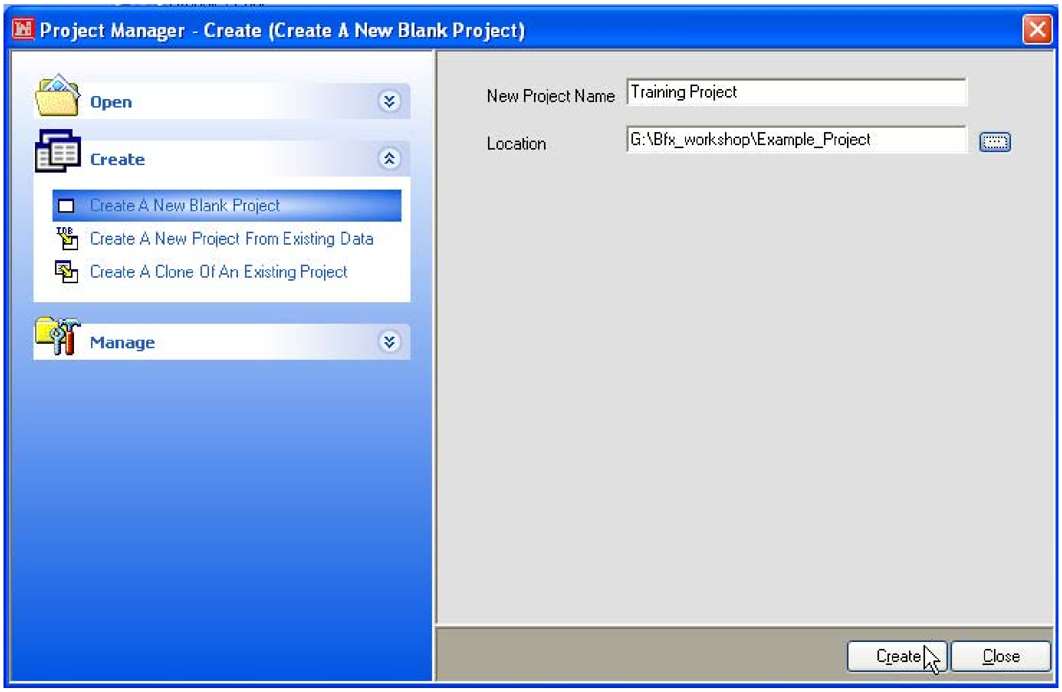
The File menu is used to open the *Project Manager*. The *Project Manager* function is to organize and manage the user's project files. The *Project Manager* utility contains three functions; **Open**, **Create**, and **Manage**. The **Open** function is used to open an existing project. Previously created projects are available from a pull-down list for selection. The **Create** function is used to create new projects. Three options are available for creating a new project; **Create a New Blank Project**, **Create a New Project From Existing Data**, and **Create a Clone of an Existing Project**. The **Manage** function provides four project/data managing capabilities; **Locate Existing Project Files**, **Delete an Existing Project**, **Rename an Existing Project**, and **Compact and Repair Beach-*fx* Databases**.

**Step 1** Open *Project Manager*, select **Create**, select **Create A New Blank Project**.



**Step 2** Enter a New Project Name **Training Project**.

**Step 3** Click the **Browse** button adjacent to the Location dialog and locate the folder where you will compile the example project.




**Step 4** Click **Create**. Reply **Yes** to the Pop-up question "Would You Like To Open The Project Now?" **Close** the *Project Manager*.

# Exercise 2
# Import Storms and Define Storm Seasons

This exercise illustrates the Beach-*fx* import capability from Microsoft Excel spreadsheets, introduces the user to the plausible storm suite data and attributes and familiarizes the user with compiling the Storm Season specification.

## 2.1 Import the Plausible Storm Suite

Beach-*fx* has the capability to import various input data into the system via Microsoft Excel spreadsheets. Templates of these import spreadsheets are provided as part of the software installation. A screen capture of a portion of the ImportStorms.xls Excel spreadsheet is shown in Figure 10.

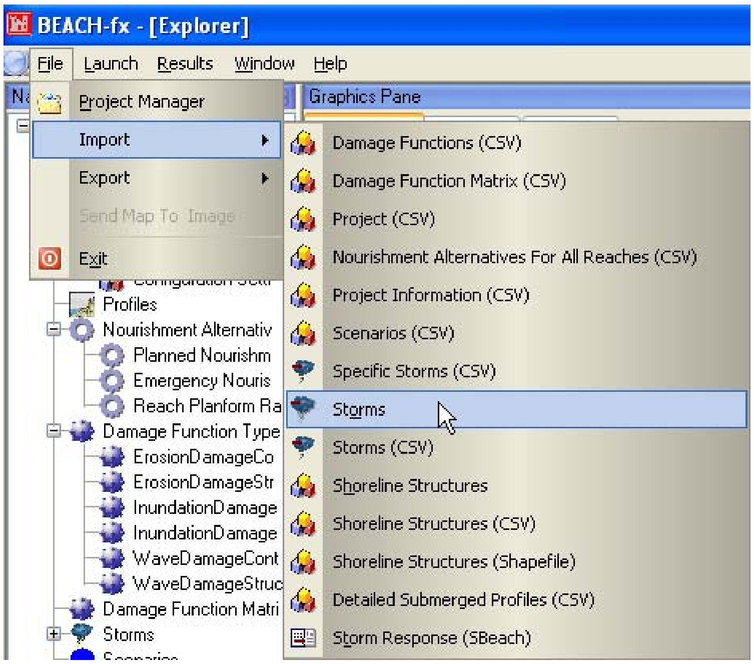
| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | StormIdentifier | StormNumber | StormType | RelativeProbability | PeakSurgePlusTide | DateOfStorm |
| 2 | WCe_18860630_H1 | 1 | 1 | 1 | 2.56 | 6/30/1886 |
| 3 | WCe_18860630_H2 | 2 | 1 | 1 | 2.07 | 6/30/1886 |
| 4 | WCe_18860630_H3 | 3 | 1 | 1 | 2.14 | 6/30/1886 |
| 5 | WCe_18860630_H4 | 4 | 1 | 1 | 2.08 | 6/30/1886 |
| 6 | WCe_18860630_L1 | 5 | 1 | 1 | 2.06 | 6/30/1886 |
| 7 | WCe_18860630_L2 | 6 | 1 | 1 | 1.64 | 6/30/1886 |
| 8 | WCe_18860630_L3 | 7 | 1 | 1 | 1.65 | 6/30/1886 |
| 9 | WCe_18860630_L4 | 8 | 1 | 1 | 1.64 | 6/30/1886 |
| 10 | WCe_18860630_M1 | 9 | 1 | 2 | 2.28 | 6/30/1886 |
| 11 | WCe_18860630_M2 | 10 | 1 | 2 | 1.79 | 6/30/1886 |
| 12 | WCe_18860630_M3 | 11 | 1 | 2 | 1.87 | 6/30/1886 |
| 13 | WCe_18860630_M4 | 12 | 1 | 2 | 1.80 | 6/30/1886 |
| 14 | WCe_18870727_H1 | 13 | 1 | 1 | 4.82 | 7/27/1887 |
| 15 | WCe_18870727_H2 | 14 | 1 | 1 | 3.92 | 7/27/1887 |
| 16 | WCe_18870727_H3 | 15 | 1 | 1 | 2.97 | 7/27/1887 |
| 17 | WCe_18870727_H4 | 16 | 1 | 1 | 3.94 | 7/27/1887 |
| 18 | WCe_18870727_L1 | 17 | 1 | 1 | 4.32 | 7/27/1887 |
| 19 | WCe_18870727_L2 | 18 | 1 | 1 | 3.90 | 7/27/1887 |
| 20 | WCe_18870727_L3 | 19 | 1 | 1 | 3.47 | 7/27/1887 |

Figure 10. ImportStorms.xls Excel import spreadsheet.

The attribute StormIdentifier is the storm name and must be identical to the storm name used in the SBEACH simulations. The attribute StormNumber is a unique sequential ID number for each of the storms in the plausible storm suite (1 through N). The attribute StormType denotes the storm type as being either Extra-Tropical (0) or Tropical (1). The attribute RelativeProbability defines the relative probability between the storms in the plausible storm suite (a storm with a relative probability of 2 is twice as likely to be selected as a storm with a relative probability of 1).

The attribute PeakSurgePlusTide is used as the elevation of flooding from the back bay. The attribute DateOfStorm is the date of occurrence of the historical storm event and is used to assign a storm to a defined Storm Season.

**Step 1** Import storms, **File** | **Import** | **Storms**.




Browse for and select the file ImportStorms.xls, it is located in the source_data folder. Click **Import**. If successful, click **OK**.


## 2.2 Define Storm Seasons

Beach-*fx* requires specification of user defined Storm Seasons. A maximum of 12 Storm Seasons may be defined for a given project. For this example project we will define a total of 7 seasons; a period of no storms between 1 December and 31 May and 6-month-long seasons for June, July, August, September, October and November. For each season, the

probability of occurrence of Extra-Tropical storms, the probability of occurrence of Tropical storms, the minimum storm arrival time in days, the probability active toggle (used primarily when calibrating Beach-*fx*, as described later), the maximum number of Extra-Tropical storms in season and the maximum number of Tropical storms in season are specified. For all specified seasons the probability of occurrence of Extra-Tropical storms should be 0, the minimum storm arrival time should be 14, the probability active toggle should be checked, the maximum number of Extra-Tropical and Tropical storms should be set to 100. The probability of occurrence of Tropical storms should be calculated based on the number of storms in the specific season divided by the total number of years covered by the plausible storm suite database. The plausible storm suite database covers a 120-year period (1886 through 2005). The number of storms in each season is listed below.

Table 1. Number of storms by season.

| Season Description | Number of Storms |
|---|---|
| No storms | 0 |
| June storms | 4 |
| July storms | 6 |
| August storms | 8 |
| September storms | 22 |
| October storms | 7 |
| November storms | 1 |

**Step 1** Click on tree item **Storm Seasons**, in the Project Information section. Click in the Description column of the data matrix and enter **No Storms**, in Start Month column use pull-down to select **December**, in Start Day enter **1**, in End Month use pull-down to select **May**, in End Day enter **31**, the Probability of both storm types is **0**, the minimum storm arrival time is **14**, probability active is checked, and the maximum number of storms in season is **100** for both storm types.

**Step 2** Complete specification of the remainder of the seasons as described above. Upon completion your data grid should look like the following.

Storm Seasons

| Number | Description | Start Month | Start Day | End Month | End Day | Probability Of E | Probability Of T | Minimum Storm A | Probability Acitve | Maximum E | Maximum |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | No Storms | December | 1 | May | 31 | 0 | 0 | 14 | ☑ | 100 | 100 |
| 2 | June Storms | June | 1 | June | 30 | 0 | 0.033333 | 14 | ☑ | 100 | 100 |
| 3 | July Storms | July | 1 | July | 31 | 0 | 0.05 | 14 | ☑ | 100 | 100 |
| 4 | August Storms | August | 1 | August | 31 | 0 | 0.066667 | 14 | ☑ | 100 | 100 |
| 5 | September Stor | September | 1 | Septembe | 30 | 0 | 0.183333 | 14 | ☑ | 100 | 100 |
| 6 | October Storms | October | 1 | October | 31 | 0 | 0.058333 | 14 | ☑ | 100 | 100 |
| 7 | November Stor | November | 1 | November | 30 | 0 | 0.008333 | 14 | ☑ | 100 | 100 |
| 0 | | January | 1 | January | 1 | 0 | 0 | 0 | ☐ | 100 | 100 |

# Exercise 3
# Define Representative Profiles

This exercise involves specification of the initial condition representative Profiles and importing the Detailed Submerged Profile.

## 3.1 Define Initial Condition Profiles

Beach-*fx* employs a simplified representation of the beach profile where key morphological features are defined by points. The simplified profile shown in Figure 11, represents a single trapezoidal dune with a horizontal berm feature seaward of the dune and a horizontal upland landward of the dune. The upland elevation is constant, as is the dune slope, berm elevation and foreshore slope. The dynamic features of the representative Profile are the upland width, dune width, dune elevation and berm width. It is essential that the Description attribute of the representative Profiles be identical to the prefix of the Profile names simulated in SBEACH.

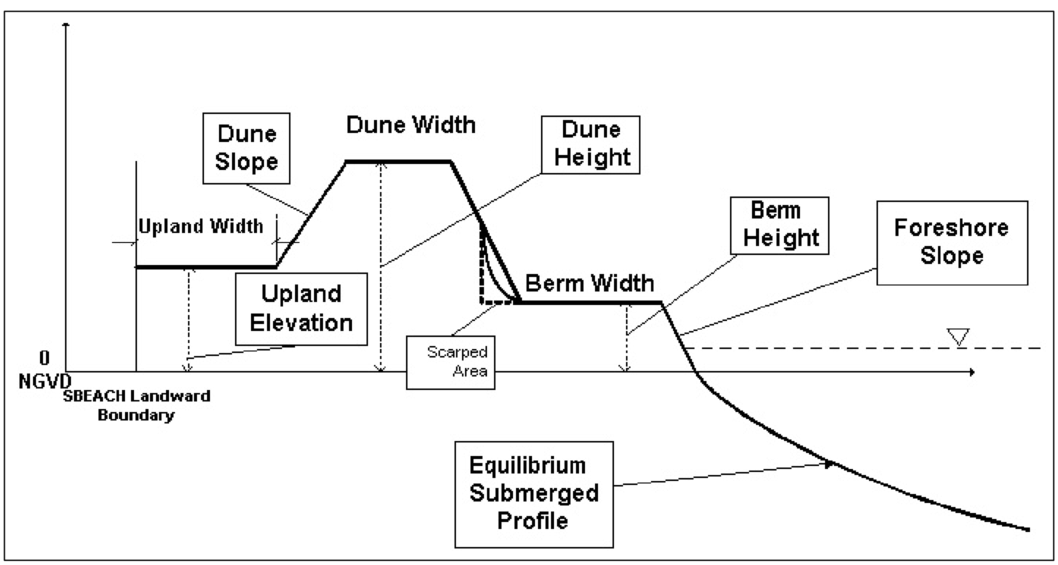



Figure 11. Beach-*fx* simplified representative beach profile.

**Step 1** Click on tree item **Profiles**, click in the Description column in the data grid and enter **R1**. Enter **7.5** for Default Dune Height, **95** for Default Dune Width, **6.6** for Default Berm Height, **0** for Default Berm Width,

**0.18** for Dune Slope, **0.0588** for Foreshore Slope, **6** for Upland Elevation, **30** for Depth of Closure, Detailed Submerged Profile Flag checked and **0** for Submerged Profile Parameter.

Click in the Description column below R1 and enter **R2R5**. Enter **16.5** for Default Dune Height, **45** for Default Dune Width, **6.6** for Default Berm Height, **0** for Default Berm Width, **0.106** for Dune Slope, **0.0588** for Foreshore Slope, **6** for Upland Elevation, **30** for Depth of Closure, Detailed Submerged Profile Flag checked and **0** for Submerged Profile Parameter.

## 3.2 Import Detailed Submerged Profiles

The submerged profile is used in Beach-*fx* for visualization purposes only and may take one of two forms; a user specified Detailed Submerged Profile or an equilibrium profile shape defined by the relationship

$$Depth = Ax^{2/3}$$

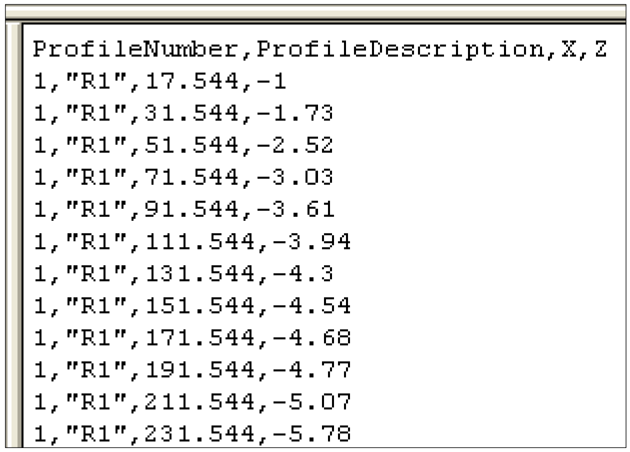
where

$A$ is the empirical sediment scale parameter

$x$ is the distance offshore from the shoreline

The Profile attribute Detailed Submerged Profile flag determines which of the two submerged profile options is activated. If this attribute is checked, then the user specifies the Detail Submerged Profile by providing distance elevation ($x$, $z$) data defining the submerged profile, where distance is distance from the shoreline. If this attribute is not checked the user specifies the value of the sediment scale parameter for the attribute Submerged Profile Parameter.

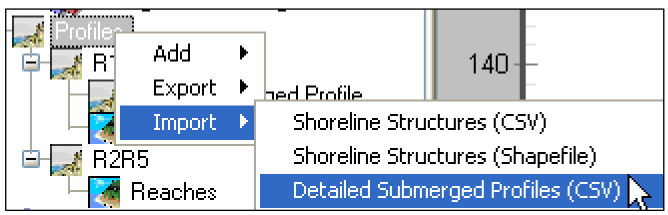
The Detailed Submerged Profile data can be imported into Beach-*fx* from a comma delimited (.csv) file as illustrated in Figure 12. Templates of the available comma delimited import files (SubmergedProfiles.csv) are provided as part of the software installation.



```
ProfileNumber,ProfileDescription,X,Z
1,"R1",17.544,-1
1,"R1",31.544,-1.73
1,"R1",51.544,-2.52
1,"R1",71.544,-3.03
1,"R1",91.544,-3.61
1,"R1",111.544,-3.94
1,"R1",131.544,-4.3
1,"R1",151.544,-4.54
1,"R1",171.544,-4.68
1,"R1",191.544,-4.77
1,"R1",211.544,-5.07
1,"R1",231.544,-5.78
```

Figure 12. SubmergedProfile.csv file.

**Step 1** Right click on tree item **Profiles** | **Import** | **Detailed Submerged Profiles (CSV)**.



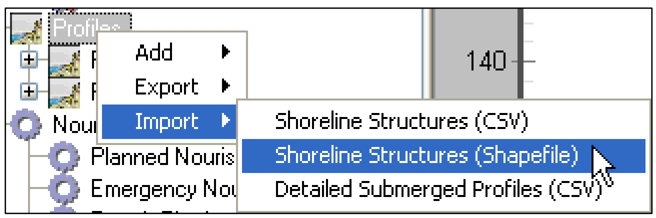

Browse for and select the file SubmergedProfiles.csv, it is located in the source_data folder. Click **Import**. If successful, click **OK**.

# Exercise 4
# Define Reaches and Lots

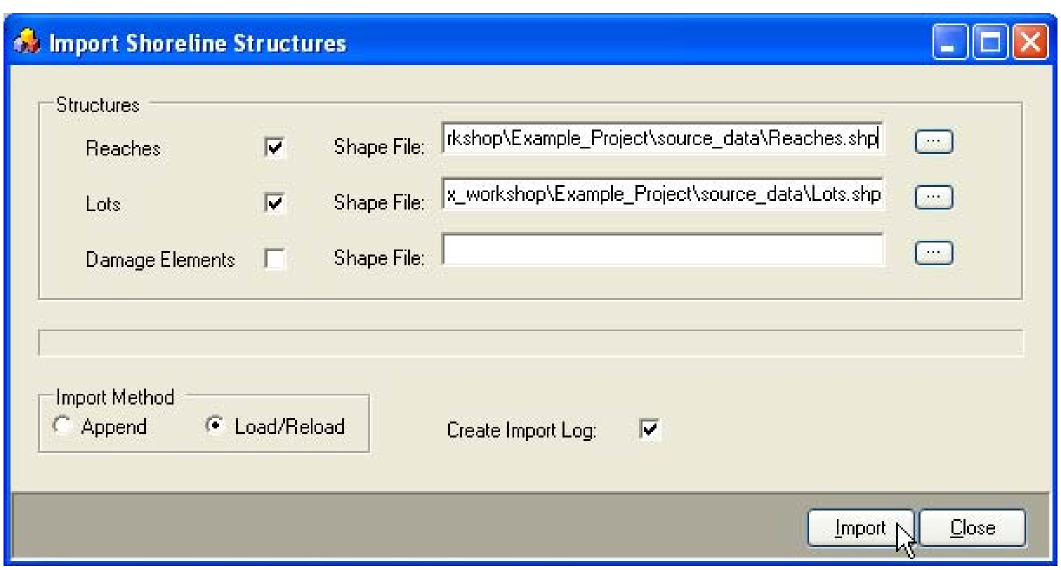
As described previously, Project Reaches and Lots can be imported into Beach-*fx* directly from template ESRI shapefiles provided as part of the software installation. This exercise involves specification of the example Project Reaches and Lots via import of previously compiled and attributed Reach and Lot shapefiles based on the Reach and Lot shapefile templates (Reaches.shp and Lots.shp).

## 4.1 Import Reaches and Lots

**Step 1** Right click on tree item **Profiles** | **Import** | **Shoreline Structures (Shapefile)**.




**Step 2** In the Import Shoreline Structures dialog check Reaches and browse for and select the file Reaches.shp, it is located in the source_data folder. Check Lots and browse for the file Lots.shp, it also is located in the source_data folder. Click **Import**.



If successful, click **OK**.

# Exercise 5
# Damage Element Variables and Import Damage Element Inventory

Upon creation of a new blank project, a number of default Damage Element Variables are created related to Armor Types, Construction Types, Damage Element Types, and Foundation Types. These data tables are located in the Project Information section of the *Navigation Tree*, specifically under the Damage Element Variables branch. In order for a Damage Element to be imported into Beach-*fx*, that Damage Element's attributes of Armor Type, Construction Type, Damage Type, and Foundation Type must be present in the Damage Element Variables data tables. Although the default variables provided are common to many projects they are by no means exhaustive and the user is free to implement other type specifications in any of these tables, as needed. Consequently, it is important to check and compare each of the mentioned data tables against the attributes of the Damage Elements for the project and add any missing data types to the Damage Element Variables tables prior to importing the Damage Element inventory. Figures 13 through 16 show the default Damage Element Variable tables.

Armor Types

| Code | Description | Number |
|---|---|---|
| Null | No Armor | 1 |
| | | 0 |

Figure 13. Default Armor Types

Construction Types

| Code | Description | Number |
|---|---|---|
| WOOD | WOOD | 1 |
| MAS | MAS | 2 |
| RMAS | RMAS | 3 |
| | | 0 |

Figure 14. Default Construction Types.

| Damage Element Types | | | | | | | |
|---|---|---|---|---|---|---|---|
| Code | Description | Number | Linear Element | Linear Element Division Length | Auto Locate Elevation | Elevation Offset | Elevation Variabilty |
| SFR1 | Single-Family Residential | 1 | ☐ | | ☐ | | |
| MFR1 | Multi-Family Residential | 2 | ☐ | | ☐ | | |
| MFR2 | MF Residential (2 Floors) | 3 | ☐ | | ☐ | | |
| MFR3 | MF Residential (3 Floors) | 4 | ☐ | | ☐ | | |
| COMM | Commercial | 5 | ☐ | | ☐ | | |
| GARAGE | Garage | 6 | ☐ | | ☐ | | |
| POOL | Pool | 7 | ☑ | 10 | ☑ | 1 | 0.5 |
| WALK | Walkway | 8 | ☑ | 20 | ☑ | 3 | 0.5 |
| DECK | Deck | 9 | ☐ | | ☐ | | |
| GAZEBO | Gazebo | 10 | ☐ | | ☐ | | |
| JACUZZI | Jacuzzi | 11 | ☐ | | ☐ | | |
| ACC-PU | Public Access | 12 | ☐ | | ☐ | | |
| | | 0 | ☐ | 0 | ☐ | 0 | 0 |

Figure 15. Default Damage Element Types.

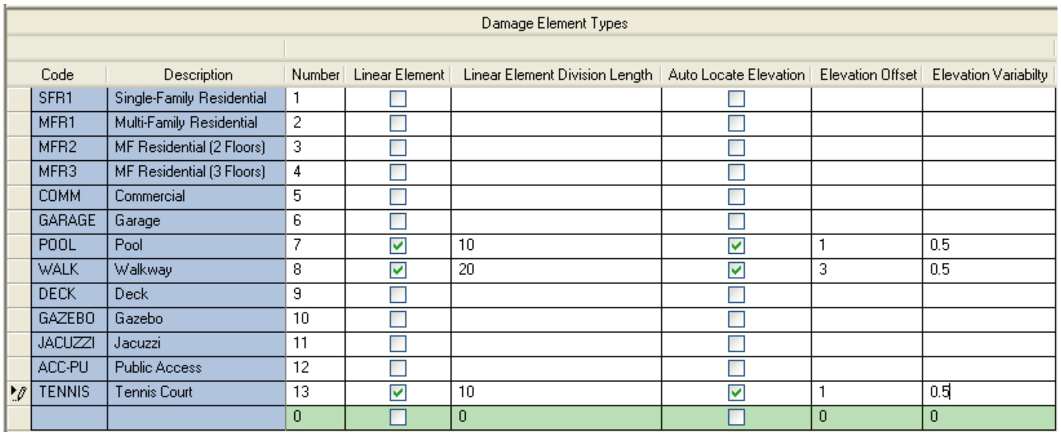
| Foundation Types | | | | |
|---|---|---|---|---|
| Code | Description | Number | Critical Erosion Amount | Shallow Foundation |
| Slab | slab on grade | 1 | 0.5 | ☑ |
| Pile | 8 foot deep piles | 2 | 4 | ☐ |
| Pile16 | 16 foot deep piles | 3 | 8 | ☐ |
| | | 0 | 0 | ☐ |

Figure 16. Default Foundation Types.

## 5.1 Check/Verify Damage Element Variables

A review of the Damage Element attributes for the example project structure inventory revealed that none of the Damage Elements are protected by armoring so, the default Armor Types specification is sufficient. The structure inventory involves only masonry (MAS) and wood (WOOD) Construction Types so, the default Construction Types specification is sufficient. The structure inventory includes seven different Damage Element Type; SFR1, MFR1, COMM, POOL, WALK, GAZEBO, and TENNIS. The default Damage Element Types include all of these except the type TENNIS. Consequently, the Damage Element Types table must be modified to add the Damage Element Type TENNIS prior to importing the Damage Element inventory. The structure inventory involves three Foundation Types; Pile, Pile16, and Slab all of which are included in the default Foundation Types specification.

**Step 1** Click on tree item **Damage Element Types**. Click in the Damage Element Types data grid in the column labeled Code (last row) and enter specifications for the Damage Element Type **TENNIS** as indicated below.

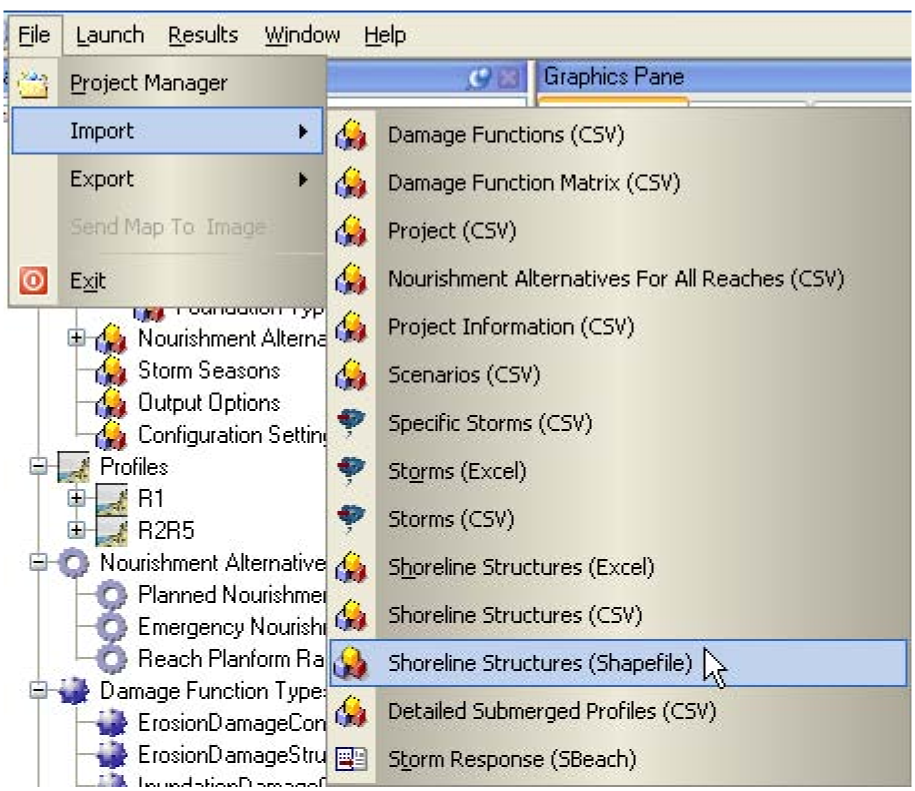


Damage Element Types

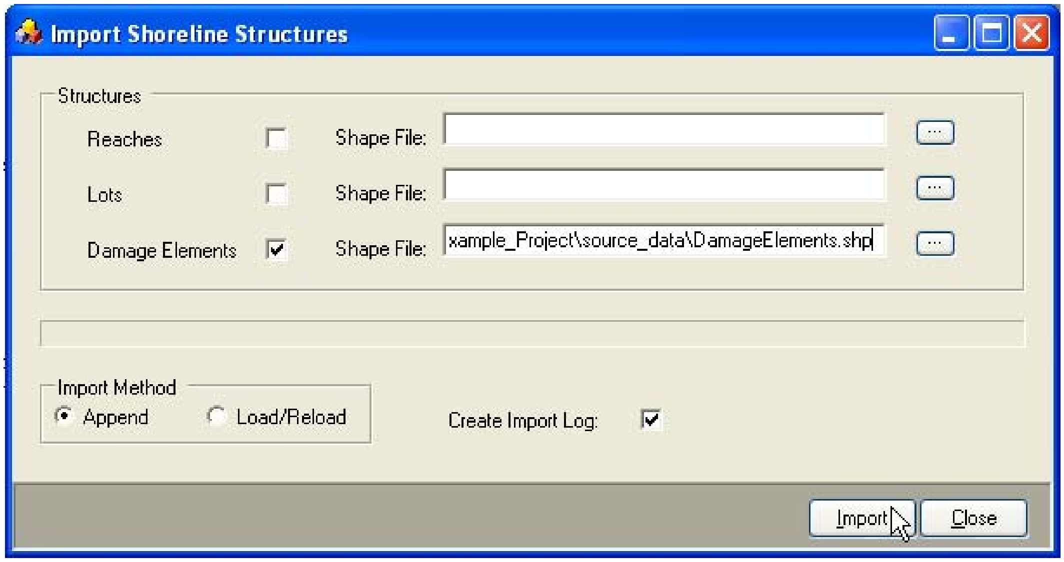
| | Code | Description | Number | Linear Element | Linear Element Division Length | Auto Locate Elevation | Elevation Offset | Elevation Variabilty |
|---|---|---|---|---|---|---|---|---|
| | SFR1 | Single-Family Residential | 1 | ☐ | | ☐ | | |
| | MFR1 | Multi-Family Residential | 2 | ☐ | | ☐ | | |
| | MFR2 | MF Residential (2 Floors) | 3 | ☐ | | ☐ | | |
| | MFR3 | MF Residential (3 Floors) | 4 | ☐ | | ☐ | | |
| | COMM | Commercial | 5 | ☐ | | ☐ | | |
| | GARAGE | Garage | 6 | ☐ | | ☐ | | |
| | POOL | Pool | 7 | ☑ | 10 | ☑ | 1 | 0.5 |
| | WALK | Walkway | 8 | ☑ | 20 | ☑ | 3 | 0.5 |
| | DECK | Deck | 9 | ☐ | | ☐ | | |
| | GAZEBO | Gazebo | 10 | ☐ | | ☐ | | |
| | JACUZZI | Jacuzzi | 11 | ☐ | | ☐ | | |
| | ACC-PU | Public Access | 12 | ☐ | | ☐ | | |
| | TENNIS | Tennis Court | 13 | ☑ | 10 | ☑ | 1 | 0.5 |
| | | | 0 | ☐ | 0 | ☐ | 0 | 0 |

## 5.2 Import Damage Element Inventory

**Step 1** Having ensured that the attributes of the Damage Element inventory are pre-defined in the Damage Element Variables tables, the Damage Element inventory can be imported from the pre-compiled shapefile template. Click **File | Import | Shoreline Structures (Shapefile)**.



In the Import Shoreline Structures dialog check Damage Elements and browse for and select the file DamageElements.shp, it is located in the source_data folder. Click **Import**.




If successful, click **OK**.

# Exercise 6
# Attach GIS Visualization

At this point, the example project information in terms of Storms, Storm Seasons, initial condition representative Profiles, Reaches, Lots, and Damage Elements have all been specified within the Beach-*fx* training project. Within the *Graphics Pane - Profile View* the relationships between the representative Profiles and Damage Elements can be visualized at multiple levels by expanding and clicking on tree items Profiles, Reaches, Lots, and Damage Elements. Clicking a specific Damage Element will cause that Damage Element to change color in the *Profile View*. Clicking on the Damage Element Symbol in the *Profile View* will open the attribute table for that specific Damage Element. Similar visualization behavior is available in the *Graphics Pane – Plan View* except in this graphics pane the data are displayed in planform as opposed to the cross-section view provided in the *Profile View*. Beach-*fx* also provides for *Map View* visualization of the project data. This *Graphics Pane* enables visualization of the GIS based data associated with the project and can include aerial photography and any available ESRI shapefiles. This exercise will illustrate the procedure for enabling project visualization within the *Map View Graphics Pane*.

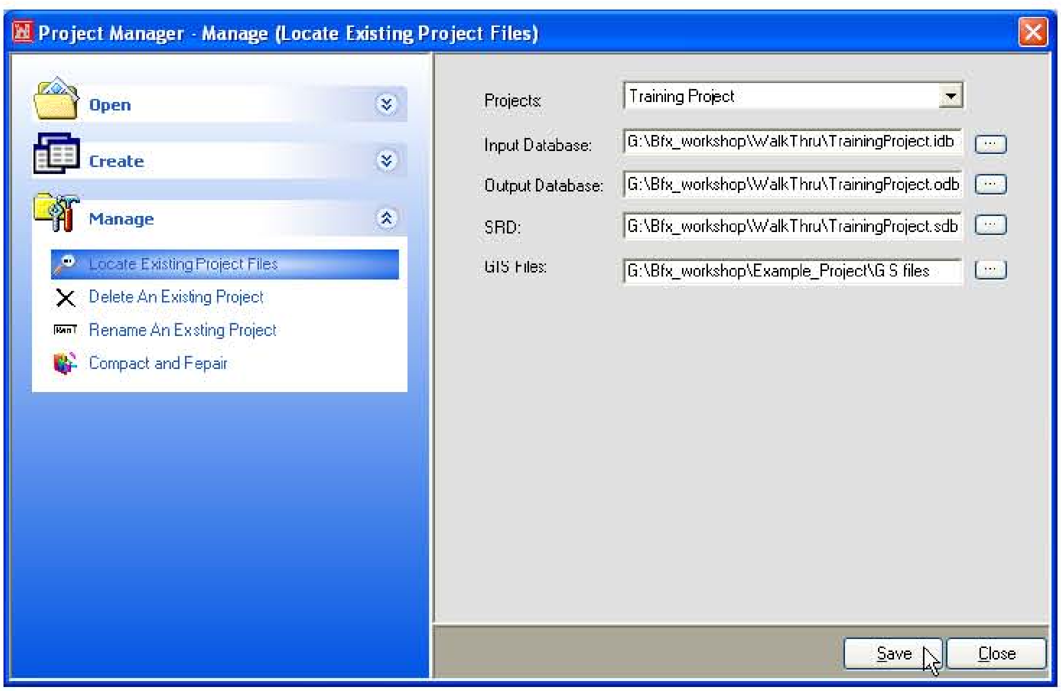
**Step 1** Using the Windows Explorer tool, create a new folder named GIS files under the Example_Project folder. Copy the files

- Example_Project.sid
- DamageElements.shp
- DamageElements.dbf
- DamageElements.shx
- Lots.shp
- Lots.dbf
- Lots.shx
- Reaches.shp
- Reaches.dbf
- Reaches.shx

to the GIS files folder.

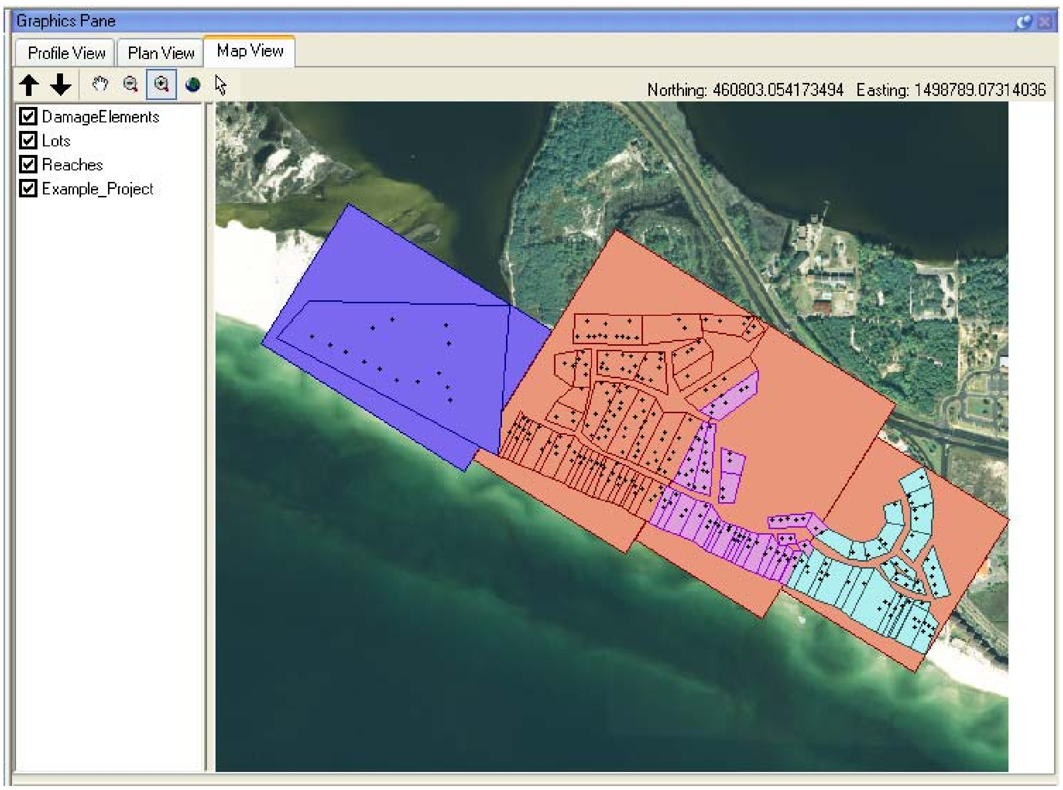
**Step 2** Within Beach-*fx* open the *Project Manager* (File | **Project Manager**). Select **Manage | Locate Existing Project Files**.


Browse for and locate the GIS files folder.




**Save** the new settings.


Close the *Project Manager*. Click on tree item **Profiles**, click on the ***Map View*** tab.




Within the *Map View*, Reaches that share the same representative Profile are shaded the same color. When a representative Profile is selected in the tree of the *Map View* automatically zooms to that area of the project and highlights the outline of that Reach. Similarly, when a specific Lot is selected in the tree, the outline of that Lot is highlighted. Likewise, when a specific Damage Element is selected in the tree, the point representing that Damage Element changes its color.

# Exercise 7
# Define Damage Functions

Beach-*fx* estimates erosion, inundation and wave impact damages to structures and their contents based on the location of the Damage Element along the SBEACH reference line (derived from the Damage Element coordinate attributes in relation to the SBEACH reference coordinates) and a damage driving parameter specific to the looked-up Profile/Storm response using user specified Damage Functions. The Damage Function provides the relationship between the damage driving parameter and the fractional loss to the structure or contents. To incorporate uncertainty each Damage Function is comprised of three curves representing minimum expected losses, most likely expected losses and maximum expected losses. Beach-*fx* draws a triangular distribution across these three curves and randomly samples within that distribution to obtain the estimated fractional loss. Damage Functions must be specified for structure and content damages for each of the three Damage Types (erosion, inundation, wave impact) for each unique combination of Damage Element Type, Foundation Type, Construction Type, and Armor Type. At a minimum, at least six Damage Functions must be specified (erosion-structures, erosion–contents, inundation–structures, inundation–contents, wave impact–structures, and wave impact–contents). However, most practical applications involve considerably more than the minimum number of Damage Functions because damages, in general, differ significantly depending on the structure's Foundation Type. Damage Functions are not assigned directly to each Damage Element. Rather, an indirect method is used, whereby the Damage Functions used for a Damage Element are determined by the type attributes of that Damage Element. Each unique combination of Armor Type, Foundation Type, Damage Element Type, and Construction Type must have six Damage Functions assigned to cover the combinations of erosion/inundation/wave impact and structure/contents. However, the same Damage Functions can be used repeatedly, that is, different combinations of Armor, Foundation, Damage Element, and Construction Type can use the same set of Damage Functions. In this example project application we employ a total of 19 Damage Functions (4 for erosion damages to contents, 4 for erosion damages to structures, 2 for inundation damages to contents, 3 for inundation damages to structures, 3 for wave impact damages to contents, and 3 for wave impact damages to structures). Damage Functions are a subject of current research and at present, there are no standard or certified Damage Functions. As such,

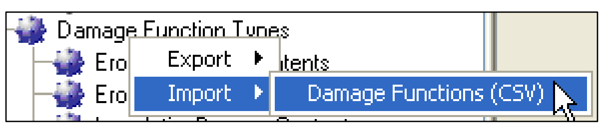
Beach-*fx* does not provide default damage functions and the user is required to specify (and be prepared to defend) Damage Dunctions used for their specific projects. Getting Damage Functions into Beach-*fx* is accomplished by first defining the Damage Function Type, name, and providing a description and then entering the Damage Function's specification in the data grid or by importing the Damage Functions from information compiled in a comma delimited (.csv) file as illustrated in Figure 17. Templates of the available comma delimited import files (DamageFunctions.csv) are provided as part of the software installation.

```
FunctionTypeName,FunctionDescriptionGroupName,FunctionDescriptionGroupDescription,X,YMin,YMostLikely,YMax
"ErosionDamageContents","EROPILECON","Erosion/Pile/Contents",0,0,0,0
"ErosionDamageContents","EROPILECON","Erosion/Pile/Contents",10,0.05,0.2,0.25
"ErosionDamageContents","EROPILECON","Erosion/Pile/Contents",20,0.06,0.4,0.6
"ErosionDamageContents","EROPILECON","Erosion/Pile/Contents",30,0.08,0.6,1
"ErosionDamageContents","EROPILECON","Erosion/Pile/Contents",40,0.1,0.8,1
"ErosionDamageContents","EROPILECON","Erosion/Pile/Contents",50,0.17,1,1
"ErosionDamageContents","EROPILECON","Erosion/Pile/Contents",60,0.32,1,1
"ErosionDamageContents","EROPILECON","Erosion/Pile/Contents",70,0.47,1,1
"ErosionDamageContents","EROPILECON","Erosion/Pile/Contents",80,0.6,1,1
"ErosionDamageContents","EROPILECON","Erosion/Pile/Contents",90,0.7,1,1
"ErosionDamageContents","EROPILECON","Erosion/Pile/Contents",100,0.8,1,1
```

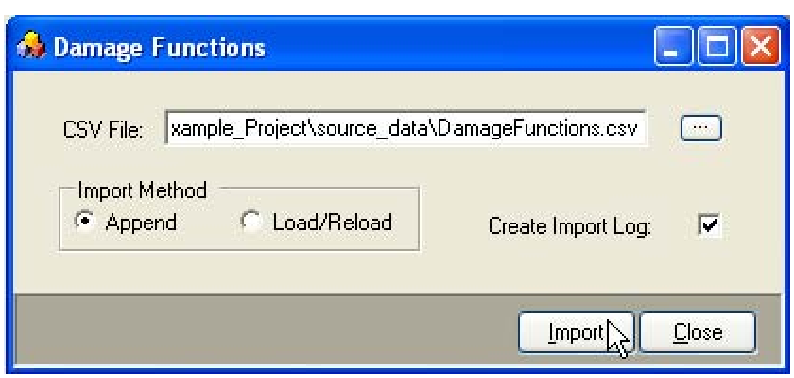
Figure 17. DamageFunctions.csv file.

## 7.1 Import Damage Element Inventory

**Step 1** Right click on the tree item **Damage Functions Types | Import | Damage Functions (CSV)**.




Browse for and select the file DamageFunctions.csv. Click **Import**.



If successful click **OK**.

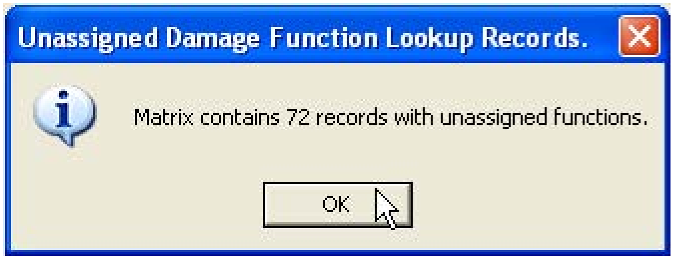
# Exercise 8
# Populate Damage Function Matrix

This exercise illustrates population of the Damage Function Matrix. The Damage Function Matrix establishes linkages between the various Damage Element Types, Foundation Types, Construction Types, damage component (structure or content), Damage Type (erosion, inundation, or wave impact), and the specific Damage Functions. Beach-*fx* automatically identifies the unique combinations and creates a matrix and the user is required to identify the Damage Function to be applied for the combination defined by each row in the matrix. In this example project, there are 72 unique combinations that require a Damage Function specification. Table 2 provides a listing of how the Damage Functions should be assigned.

Table 2. Damage function relationships.

| Damage Function | Damage Type | DE Type | Damage Component | Foundation Type |
|---|---|---|---|---|
| EROSHLSTR | Erosion | all | Structure | Slab |
| EROPILESTR | Erosion | all | Structure | Pile |
| ERODP1SSTR | Erosion | SFR1, COMM | Structure | Pile16 |
| ERODP1MSTR | Erosion | MFR1 | Structure | Pile16 |
| EROSHLCON | Erosion | all | Content | Slab |
| EROPILECON | Erosion | all | Content | Pile |
| ERODP1SCON | Erosion | SFR1, COMM | Content | Pile16 |
| ERODP1MCON | Erosion | MFR1 | Content | Pile16 |
| IWFNP | Inundation | SFR1 | Structure | Slab |
| IWFP | Inundation | SFR1, MFR1, COMM | Structure | Pile, Pile16 |
| NOFUNCTSPECIS | Inundation | POOL, TENNIS, WALK, GAZEBO | Structure | all |
| 1SNBC | Inundation | SFR1, MFR1, COMM | Content | all |
| NOFUNCTSPECIC | Inundation | POOL, TENNIS, WALK, GAZEBO | Content | all |
| WAVENPS | Wave | SFR1 | Structure | Slab |
| WAVEPS | Wave | all | Structure | Pile, Pile16 |
| NOFUNCTSPECWS | Wave | POOL, TENNIS | Structure | all |
| WAVENPC | Wave | SFR1 | Content | Slab |
| WAVEPC | Wave | all | Content | Pile, Pile16 |
| NOFUNCTSPECWC | Wave | POOL, TENNIS | Content | all |

**Step 1** Select **Damage Function Matrix** on the *Navigation Tree*. Beach-*fx* automatically checks the matrix for unassigned Damage Functions and reports that 72 records lack Damage Function assignments. Click **OK** to proceed.




**Step 2** Place the cursor in the green column labeled Function and select the appropriate damage function from the pull-down menu for that row using the information in Table 2 as a guide. Note that the attribute table can be filtered to isolate specific attribute combinations listed in the Table 2. Upon completion, the Damage Function assignments should be as follows:

Damage Function Lookups

| Function | Type | Damage Element Ty | Damage Compon | Foundation Ty | Construction Ty | Armor Typ |
|---|---|---|---|---|---|---|
| ERODP1SCON | Erosion | SFR1 | Contents | Pile16 | WOOD | No Armor |
| ERODP1MCON | Erosion | MFR1 | Contents | Pile16 | MAS | No Armor |
| ERODP1SCON | Erosion | SFR1 | Contents | Pile16 | MAS | No Armor |
| ERODP1SCON | Erosion | COMM | Contents | Pile16 | WOOD | No Armor |
| EROPILECON | Erosion | SFR1 | Contents | Pile | WOOD | No Armor |
| EROPILECON | Erosion | SFR1 | Contents | Pile | MAS | No Armor |
| EROPILECON | Erosion | WALK | Contents | Pile | WOOD | No Armor |
| EROPILECON | Erosion | GAZEBO | Contents | Pile | WOOD | No Armor |
| EROSHLCON | Erosion | SFR1 | Contents | Slab | WOOD | No Armor |
| EROSHLCON | Erosion | SFR1 | Contents | Slab | MAS | No Armor |
| EROSHLCON | Erosion | POOL | Contents | Slab | MAS | No Armor |
| EROSHLCON | Erosion | TENNIS | Contents | Slab | MAS | No Armor |
| ERODP1MSTR | Erosion | MFR1 | Structure | Pile16 | MAS | No Armor |
| ERODP1SSTR | Erosion | SFR1 | Structure | Pile16 | WOOD | No Armor |
| ERODP1SSTR | Erosion | SFR1 | Structure | Pile16 | MAS | No Armor |
| ERODP1SSTR | Erosion | COMM | Structure | Pile16 | WOOD | No Armor |
| EROPILESTR | Erosion | SFR1 | Structure | Pile | WOOD | No Armor |
| EROPILESTR | Erosion | SFR1 | Structure | Pile | MAS | No Armor |
| EROPILESTR | Erosion | WALK | Structure | Pile | WOOD | No Armor |
| EROPILESTR | Erosion | GAZEBO | Structure | Pile | WOOD | No Armor |
| EROSHLSTR | Erosion | SFR1 | Structure | Slab | WOOD | No Armor |
| EROSHLSTR | Erosion | SFR1 | Structure | Slab | MAS | No Armor |
| EROSHLSTR | Erosion | POOL | Structure | Slab | MAS | No Armor |
| EROSHLSTR | Erosion | TENNIS | Structure | Slab | MAS | No Armor |

Erosion Damage Function Assignments

| Damage Function Lookups | | | | | | |
|---|---|---|---|---|---|---|
| Function | Type | Damage Element | Damage Comp | Foundation T | Construction | Armor T |
| 1SNBC | Inundatio | SFR1 | Contents | Slab | WOOD | No Armor |
| 1SNBC | Inundatio | SFR1 | Contents | Pile | WOOD | No Armor |
| 1SNBC | Inundatio | SFR1 | Contents | Pile16 | WOOD | No Armor |
| 1SNBC | Inundatio | SFR1 | Contents | Slab | MAS | No Armor |
| 1SNBC | Inundatio | SFR1 | Contents | Pile | MAS | No Armor |
| 1SNBC | Inundatio | SFR1 | Contents | Pile16 | MAS | No Armor |
| 1SNBC | Inundatio | MFR1 | Contents | Pile16 | MAS | No Armor |
| 1SNBC | Inundatio | COMM | Contents | Pile16 | WOOD | No Armor |
| NOFUNCTSPECIC | Inundatio | POOL | Contents | Slab | MAS | No Armor |
| NOFUNCTSPECIC | Inundatio | WALK | Contents | Pile | WOOD | No Armor |
| NOFUNCTSPECIC | Inundatio | GAZEBO | Contents | Pile | WOOD | No Armor |
| NOFUNCTSPECIC | Inundatio | TENNIS | Contents | Slab | MAS | No Armor |
| IWFNP | Inundatio | SFR1 | Structure | Slab | WOOD | No Armor |
| IWFNP | Inundatio | SFR1 | Structure | Slab | MAS | No Armor |
| IWFP | Inundatio | SFR1 | Structure | Pile | WOOD | No Armor |
| IWFP | Inundatio | SFR1 | Structure | Pile16 | WOOD | No Armor |
| IWFP | Inundatio | SFR1 | Structure | Pile | MAS | No Armor |
| IWFP | Inundatio | SFR1 | Structure | Pile16 | MAS | No Armor |
| IWFP | Inundatio | MFR1 | Structure | Pile16 | MAS | No Armor |
| IWFP | Inundatio | COMM | Structure | Pile16 | WOOD | No Armor |
| NOFUNCTSPECIS | Inundatio | POOL | Structure | Slab | MAS | No Armor |
| NOFUNCTSPECIS | Inundatio | WALK | Structure | Pile | WOOD | No Armor |
| NOFUNCTSPECIS | Inundatio | GAZEBO | Structure | Pile | WOOD | No Armor |
| NOFUNCTSPECIS | Inundatio | TENNIS | Structure | Slab | MAS | No Armor |

Inundation Damage Function Assignments

Damage Function Lookups

| Function | Type | Damage Element | Damage Comp | Foundation T | Construction | Armor T |
|---|---|---|---|---|---|---|
| NOFUNCTSPECWC | Wave | POOL | Contents | Slab | MAS | No Armor |
| NOFUNCTSPECWC | Wave | TENNIS | Contents | Slab | MAS | No Armor |
| WAVENPC | Wave | SFR1 | Contents | Slab | WOOD | No Armor |
| WAVENPC | Wave | SFR1 | Contents | Slab | MAS | No Armor |
| WAVEPC | Wave | SFR1 | Contents | Pile | WOOD | No Armor |
| WAVEPC | Wave | SFR1 | Contents | Pile16 | WOOD | No Armor |
| WAVEPC | Wave | SFR1 | Contents | Pile | MAS | No Armor |
| WAVEPC | Wave | SFR1 | Contents | Pile16 | MAS | No Armor |
| WAVEPC | Wave | MFR1 | Contents | Pile16 | MAS | No Armor |
| WAVEPC | Wave | WALK | Contents | Pile | WOOD | No Armor |
| WAVEPC | Wave | COMM | Contents | Pile16 | WOOD | No Armor |
| WAVEPC | Wave | GAZEBO | Contents | Pile | WOOD | No Armor |
| NOFUNCTSPECWS | Wave | POOL | Structure | Slab | MAS | No Armor |
| NOFUNCTSPECWS | Wave | TENNIS | Structure | Slab | MAS | No Armor |
| WAVENPS | Wave | SFR1 | Structure | Slab | WOOD | No Armor |
| WAVENPS | Wave | SFR1 | Structure | Slab | MAS | No Armor |
| WAVEPS | Wave | SFR1 | Structure | Pile | WOOD | No Armor |
| WAVEPS | Wave | SFR1 | Structure | Pile16 | WOOD | No Armor |
| WAVEPS | Wave | SFR1 | Structure | Pile | MAS | No Armor |
| WAVEPS | Wave | SFR1 | Structure | Pile16 | MAS | No Armor |
| WAVEPS | Wave | MFR1 | Structure | Pile16 | MAS | No Armor |
| WAVEPS | Wave | WALK | Structure | Pile | WOOD | No Armor |
| WAVEPS | Wave | COMM | Structure | Pile16 | WOOD | No Armor |
| WAVEPS | Wave | GAZEBO | Structure | Pile | WOOD | No Armor |

Wave Damage Function Assignments

# Exercise 9
# Build/Populate Shore Response Database

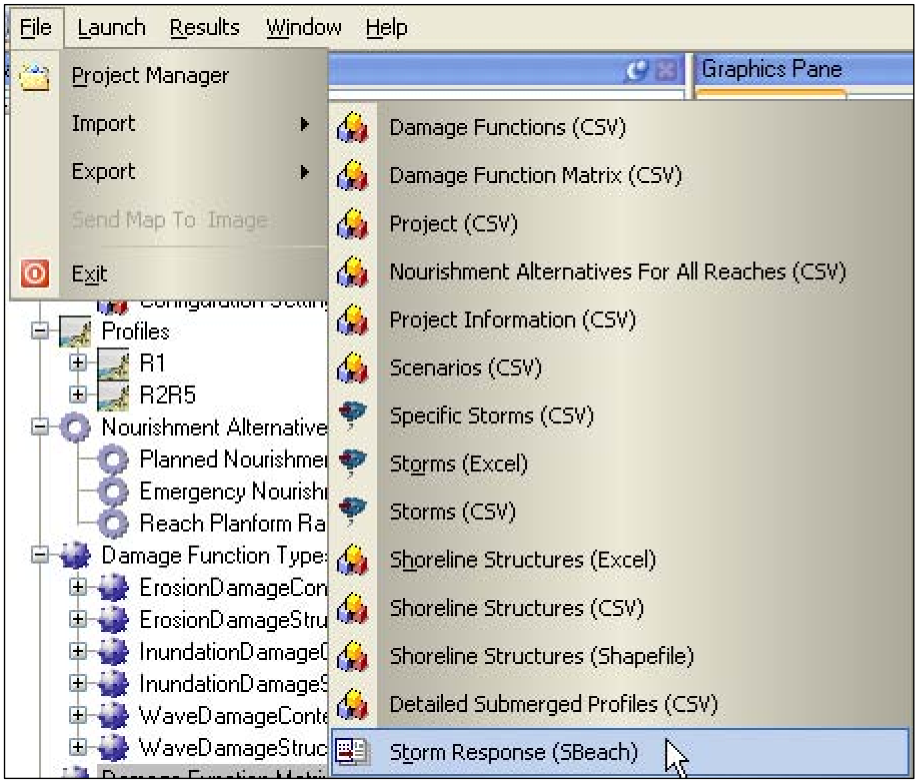
The last required step for development of the example project within Beach *fx* is population of the Shore Response Database (SRD). This exercise illustrates the functionality of the SBEACH data extractor. After the engineering activity of performing the required SBEACH simulations has been completed and the results have been quality checked, the SBEACH files are processed as follows:

- Within SBEACH perform a Global Export (name the output file with a .dat extension).
- Collect the Global Export .dat files to be processed in a single folder.
- Within Beach-*fx* execute the SBEACH data extractor. The data extractor will examine the detailed SBEACH-created ASCII format .dat files and process them to obtain the simplified profile representations and cross-shore erosion, inundation and wave damage profiles and insert them into the Shore Response Database.

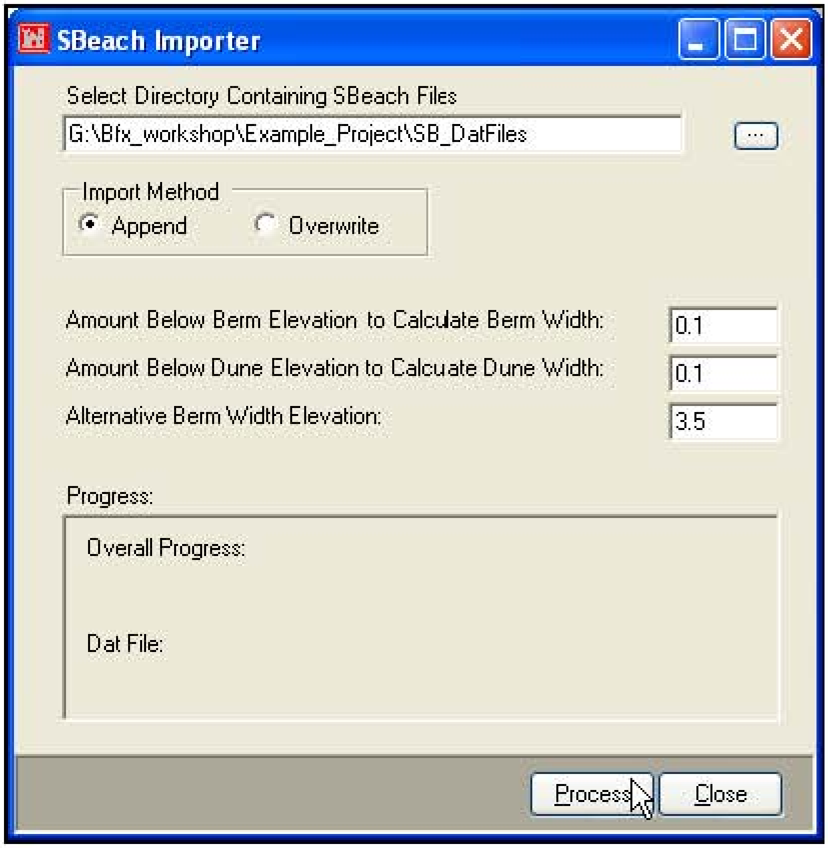
Because of the computational effort (and associated wall clock duration) of this activity, this exercise will involve processing of a single Global Export file from SBEACH containing just 68 simulations (Profile/Storm combinations).

**Step 1** Create a folder named SB_DatFiles under the Example Project folder and copy the file SB_GlobalExport.dat into this folder.

**Step 2** In Beach-*fx* click **File | Import | Storm Response (SBEACH)**.




**Step 3** In the SBEACH Importer dialog browse select the folder created in Step 1. Set the “Amount Below Berm Elevation to Calculate Berm Width” to **0.1**. Set the “Amount Below Dune Elevation to Calculate Dune Width” to **0.1**. Leave “Alternative Berm Width Elevation” at **3.5**. Click **Process**, to initiate population of the SRD.


SBeach Importer
Select Directory Containing SBeach Files
G:\Bfx_workshop\Example_Project\SB_DatFiles
...
Import Method
Append
Overwrite
Amount Below Berm Elevation to Calculate Berm Width:
0.1
Amount Below Dune Elevation to Calcuate Dune Width:
0.1
Alternative Berm Width Elevation:
3.5
Progress:
Overall Progress:
Dat File:
Process
Close

# Training Exercises

# Part II

# Beach-*fx* Calibration Procedure

# Introduction to Beach-*fx* Calibration Process

Calibration of Beach-*fx* is achieved through an iterative simulation process in which a balance is reached between three interrelated model specifications and is based on the concept that in the absence of management activities the simulated rate of shoreline change should, on average over multiple life-cycles, equal the estimated historical rate of shoreline change. Calibration is essential to insure that the morphology behavior is appropriate and representative of the situation in the study area. Calibration must be carried out prior to using Beach-*fx* for any economic calculations.

The overall rate of shoreline change in Beach-*fx* is a function of:

- Storm Climatology
- Post-Storm Berm Width Recovery
- Applied Erosion Rate

The goal, therefore, is to determine, on a Reach-by-Reach basis, the combination of specifications that will cause Beach-*fx* to return, on average over multiple life-cycles, the target historical erosion.

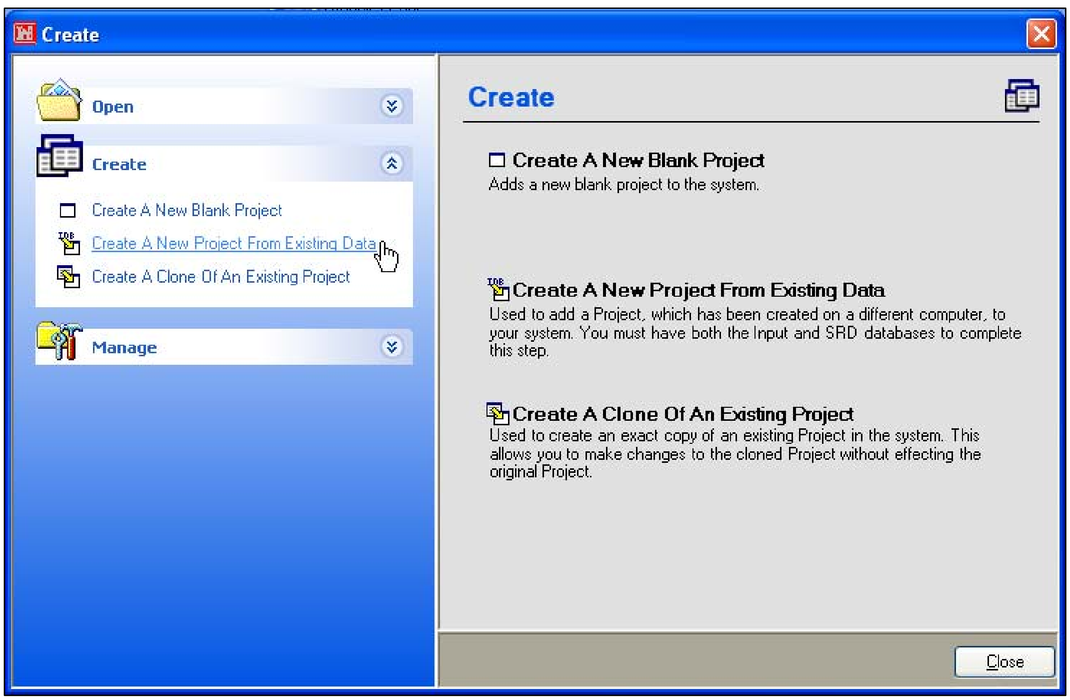
The calibration procedure involves two preliminary steps involving one simulation each. The final step involves a series of simulations. The first step is to confirm the role of the applied erosion rate (Exercise 10). The role of the applied erosion rate is isolated by making a simulation in which there are no storms and the only process causing shoreline change is that induced by the applied erosion rate. The second step is to determine the combined effect of the storm climatology and the specified post-storm berm width recovery on the estimated rate of shoreline change in the absence of an applied erosion rate (Exercise 11). This step identifies the general magnitude of shoreline rate of change due to storm processes. The third and final step is to determine the values of the applied erosion rate, on a Reach-by-Reach basis that will cause Beach-*fx* to return the target historical erosion rate (Exercise 12). The applied erosion rate can be thought of as the expected rate of shoreline change in the absence of storms at the project study area.

As a preliminary to performing the Beach-*fx* calibration exercises, create a new project named TrainingProjectCal using the files:

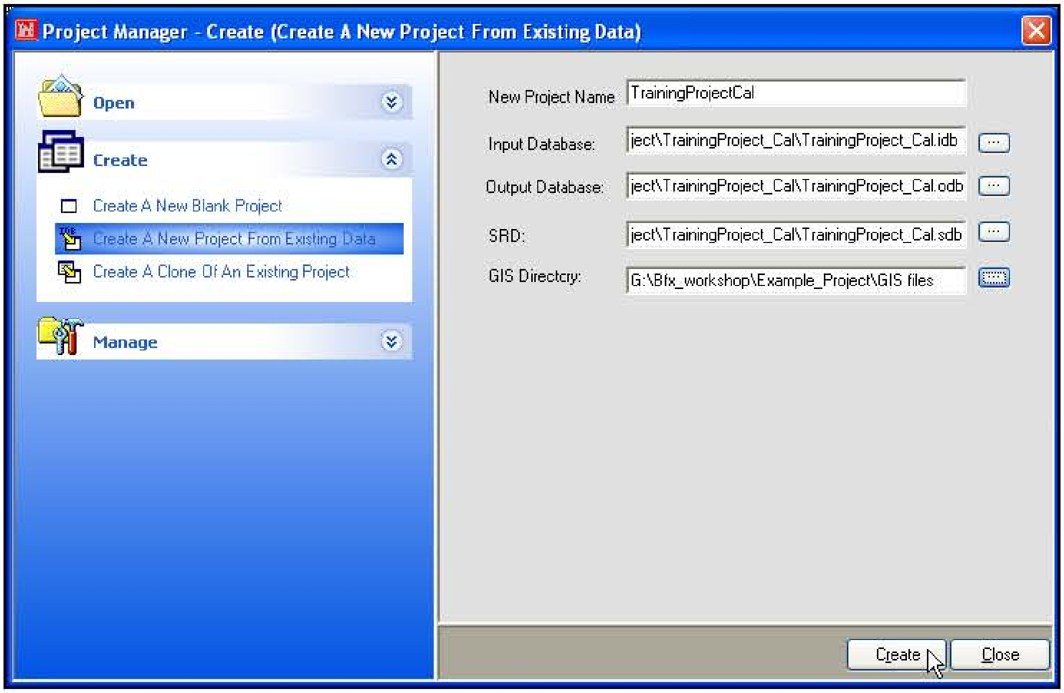
TrainingProject_Cal.idb
TrainingProject_Cal.odb
TrainingProject_Cal.sdb

**Step 1** Using the Windows Explorer Tool create a new folder named **TrainingProject_Cal** under the folder Example_Project and copy TrainingProject_Cal.idb, TrainingProject_Cal.odb and TrainingProject_Cal.sdb into the new folder.

**Step 2** In Beach-*fx* click **File** and activate the *Project Manager*. Select **Create A New Project From Existing Data**.




**Step 3** Name the new project **ProjectTrainingCal**. Browse for and select the TrainingProject_Cal databases and point to the previously created GIS folder. Click **Create**. After project is opened click the **Close** button.


Project Manager - Create (Create A New Project From Existing Data)
Open
Create
Create A New Blank Project
Create A New Project From Existing Data
Create A Clone Of An Existing Project
Manage
New Project Name
TrainingProjectCal
Input Database:
ject\TrainingProject_Cal\TrainingProject_Cal.idb
Output Database:
ject\TrainingProject_Cal\TrainingProject_Cal.odb
SRD:
ject\TrainingProject_Cal\TrainingProject_Cal.sdb
GIS Directory:
G:\Bfx_workshop\Example_Project\GIS files
Create
Close

# Exercise 10
# Role of ‘Applied Erosion Rate’ (AER)

**Step 1** Set all Storm Seasons inactive. Click *Navigation Tree* item **Storm Seasons**, in the data grid deactivate all Storm Seasons by clicking the **Probability Active** toggle box such that the box is empty (as opposed to containing a check mark). This ensures that no storms will be generated for the Beach-*fx* Simulation.

Storm Seasons

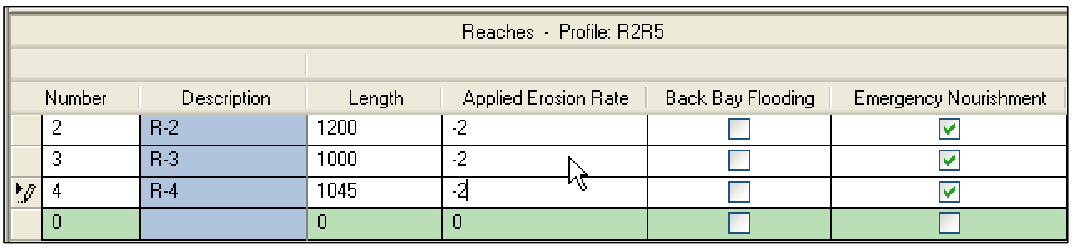
| Number | Description | n | Minimum Storm Arrival Time | Probability Acitve | Maximum Extra-Tropical Storms |
|---|---|---|---|---|---|
| 1 | No Storms | | 14 | ☐ | 100 |
| 2 | June Storms | | 14 | ☐ | 100 |
| 3 | July Storms | | 14 | ☐ | 100 |
| 4 | August Storms | | 14 | ☑ | 100 |
| 5 | September Storm | | 14 | ☑ | 100 |
| 6 | October Storms | | 14 | ☑ | 100 |
| 7 | November Storms | | 14 | ☑ | 100 |
| 0 | | | 0 | ☐ | 100 |

**Step 2** Click *Navigation Tree* item Profiles, in the data grid column Default Berm Width. Assign a default berm width of **100** for both representative Profiles. The reason for changing the default berm width from 0 to 100 is to provide sufficient berm width for the applied erosion rate to act directly on the berm without interference due to dune scarping and related volumetric calculations which will influence the calculated shoreline rate of change.

Profiles

| Number | Description | Default Dune Height | Default Dune Width | Default Berm Height | Default Berm Width |
|---|---|---|---|---|---|
| 1 | R1 | 7.5 | 95 | 6.6 | 100 |
| 2 | R2R5 | 16.5 | 45 | 6.6 | 100 |
| 0 | | 0 | 0 | 0 | 0 |

**Step 3** Assign an Applied Erosion Rate of -2 ft/year to each of the four Reaches. Expand representative Profile R1, expand Reaches, click Reach **R-1**, change Reach attribute Applied Erosion Rate to -2. Expand representative Profile R2R5, click **Reaches**, change Reach attribute Applied Erosion Rate to **-2** in Reaches R -2, R-3, and R-4.


Reaches - Profile: R2R5

| Number | Description | Length | Applied Erosion Rate | Back Bay Flooding | Emergency Nourishment |
|---|---|---|---|---|---|
| 2 | R-2 | 1200 | -2 | ☐ | ☑ |
| 3 | R-3 | 1000 | -2 | ☐ | ☑ |
| 4 | R-4 | 1045 | -2 | ☐ | ☑ |
| 0 | | 0 | 0 | ☐ | ☐ |

**Step 4** Right click *Navigation Tree* item **Scenarios** | **Add** | **New Scenario**. This will create a new simulation Scenario.


Click the Scenario **New Scenario**, in the data area complete and the dialog form:

Name – AER, Description – Confirm Applied Erosion Rate,

**Scenario Data**

Iterations – 1, Duration – 50, Interest – 0.0525,

Start Year – 2000, Start Month – 1, Base Year – 2000,

Seed – 15486583, Emergency Nourishment – off,

Start in step mode – off, Handle Scarping – on,

Calibration Run – on.

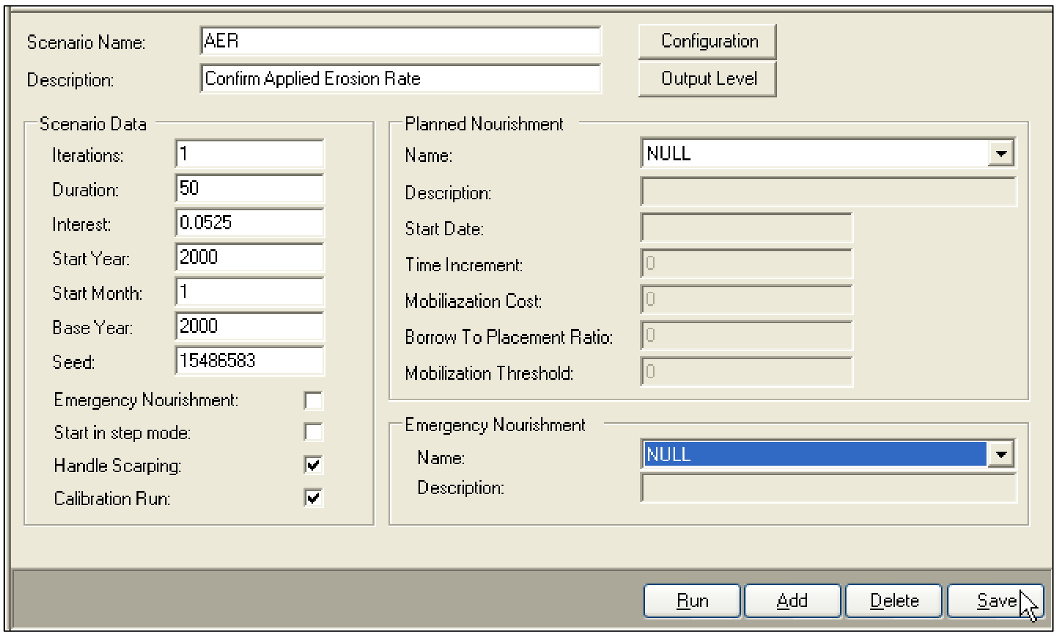
**Planned Nourishment**

Select **NULL** from the pull-down.

**Emergency Nourishment**

Select **NULL** from the pull-down.

Note that it is important to check the “calibration run” checkbox for all calibration runs—this will greatly speed up the simulation, because economic processing is not carried out in the calibration mode.




Click **Save**

**Step 5** Click the **Configuration** button on the Scenario form. Review Configuration Settings. Set DepthOfClosureMultiplier to one (**1**).

Configuration Settings

| | Tag | Description | Value |
|---|---|---|---|
| | BarrowFloodingFlag | Use Barrow Flooding Approach if > 0 | -1 |
| | BarrowFloodingReductionAmount | Amount to reduce the returned water level in Barrow flooding approach | 0 |
| | BarrowFloodingReductionMultiplier | Amount to multiply the returned water level in Barrow flooding approach | 1 |
| | BarrowFloodingReductionThreshold | If adjusted Barrow value < Threshold then set flooding value to 0 | 0 |
| | CondemnationRatio | global condemnation ratio | 0.5 |
| | DamageElementDistanceTolerance | Tolerance (feet) for damage element distance checks | 10 |
| | DataCheckMaximumLotSize | Maximum Lot Size, in acres, for data check flag (lots above this size will be f | 0.1 |
| | DataCheckMinimumLotSize | Minimum Lot Size, in acres, for data check flag (lots below this size will be fl | 0.1 |
| ▶ | DepthOfClosureMultiplier | Multiplier on entered depth of closure for use in morphology change calculati | 1 |
| | EmergencyNourishmentFillDensitySp | Cubic Yards/Foot Fill Density For Emergency Nourishment | 4 |
| | EmergencyNourishmentMobilizationTi | Time Window after end of last Emergency Nourishment at which mobilizatio | 20 |
| | EmergencyNourishmentScheduledNo | Multiplier on reach emegency nourishment mobilization to set blackout wind | 2 |
| | EmergencyNourishmentTriggerAdjust | Emergency Nourishment Trigger Adjustment > 0 set trigger greater than dun | 1 |
| | LookupBermWidthExceptionReportTo | Value in feet for flagging exception report on lookups for berm width | 20 |
| | LookupDuneHeightExceptionReportT | Value in feet for flagging exception report on lookups for dune height | 20 |
| | LookupDuneWidthExceptionReportTo | Value in feet for flagging exception report on lookups for dune width | 20 |
| | LotArmoringInEffect | Lot Armoring In Effect if > 0 | -1 |
| | MaximumRecoverableScarping | Maximum amount of scarping that can be recovered (feet) | 5 |
| | NoWriteToDataBase | Turn off writing to output data base if > 0 (used for testing) | 0 |
| | StormGenerationOnly | Storm Generation Run Only if > 0 | 0 |
| | StormRecoveryPeriod | Time in days after storm for recovery to be applied | 21 |
| | TimeStep | Time step for planform / historical adjustments, days | 7 |
| | TimeStepReportInterval | Report every nth time step (0 = no report, 1 = every time step, 2= every 2nd | 10 |
| | ToleranceNoProfileChange | tolerance for assume no profile change (de minimis) | 0.1 |
| | UsePlanform | if >=0 apply stored planform rates, if < 0 planform = 0 all reaches | 1 |

| Configuration Settings | | |
|---|---|---|
| Tag | Description | Value |
| BarrowFloodingFlag | Use Barrow Flooding Approach if > 0 | -1 |
| BarrowFloodingReductionAmount | Amount to reduce the returned water level in Barrow flooding approach | 0 |
| BarrowFloodingReductionMultiplier | Amount to multiply the returned water level in Barrow flooding approach | 1 |
| BarrowFloodingReductionThreshold | If adjusted Barrow value < Threshold then set flooding value to 0 | 0 |
| CondemnationRatio | global condemnation ratio | 0.5 |
| DamageElementDistanceTolerance | Tolerance (feet) for damage element distance checks | 10 |
| DataCheckMaximumLotSize | Maximum Lot Size, in acres, for data check flag (lots above this size will be f | 0.1 |
| DataCheckMinimumLotSize | Minimum Lot Size, in acres, for data check flag (lots below this size will be fl | 0.1 |
| DepthOfClosureMultiplier | Multiplier on entered depth of closure for use in morphology change calculati | 1 |
| EmergencyNourishmentFillDensitySp | Cubic Yards/Foot Fill Density For Emergency Nourishment | 4 |
| EmergencyNourishmentMobilizationTi | Time Window after end of last Emergency Nourishment at which mobilizatio | 20 |
| EmergencyNourishmentScheduledNo | Multiplier on reach emegency nourishment mobilization to set blackout wind | 2 |
| EmergencyNourishmentTriggerAdjust | Emergency Nourishment Trigger Adjustment > 0 set trigger greater than dun | 1 |
| LookupBermWidthExceptionReportTo | Value in feet for flagging exception report on lookups for berm width | 20 |
| LookupDuneHeightExceptionReportT | Value in feet for flagging exception report on lookups for dune height | 20 |
| LookupDuneWidthExceptionReportTo | Value in feet for flagging exception report on lookups for dune width | 20 |
| LotArmoringInEffect | Lot Armoring In Effect if > 0 | -1 |
| MaximumRecoverableScarping | Maximum amount of scarping that can be recovered (feet) | 5 |
| NoWriteToDataBase | Turn off writing to output data base if > 0 (used for testing) | 0 |
| StormGenerationOnly | Storm Generation Run Only if > 0 | 0 |
| StormRecoveryPeriod | Time in days after storm for recovery to be applied | 21 |
| TimeStep | Time step for planform / historical adjustments, days | 7 |
| TimeStepReportInterval | Report every nth time step (0 = no report, 1 = every time step, 2= every 2nd | 10 |
| ToleranceNoProfileChange | tolerance for assume no profile change (de minimis) | 0.1 |
| UsePlanform | if >=0 apply stored planform rates, if < 0 planform = 0 all reaches | 1 |

**Step 6** Review Output Level/Output Options. Check/Activate output for MorphologyTimeLine, ReachIteration, StormGeneration and StormIteration.

Ouput Level Settings

| Output Level Control | | | | |
|---|---|---|---|---|
| Tag | OutputControlFlag | SortOrder | Description | FileSuffix |
| StormResponse | ☐ | 1 | Pre and post-storm morphology information | _StormResponse.csv |
| MorphologyTimeLine | ☑ | 2 | Post-processing 2-D animation driver files | _MorphologyTimeLine and |
| Nourishment | ☐ | 3 | Report of costs, volume for planned and emerge | Nourishment.csv |
| ReachIteration | ☑ | 5 | Totals by reach for damages, emergency nouris | _ReachIteration.csv |
| PlannedNourishment | ☐ | 10 | Information on Each Planned Nourishment Cycl | _PlannedNourishment.csv |
| ReachPlannedNourishment | ☐ | 10 | Information on Each Reach Planned Nourishme | _PlannedNourishmentReac |
| StormGeneration | ☑ | 10 | Information on each generated storm | _StormGeneration.csv |
| StormIteration | ☑ | 10 | Statistics on number of storms by iteration, seas | _StormIteration.csv |
| CoordinateChecks | ☐ | 20 | Data Checking for Reach / Lot / Damage Eleme | _CoordinateChecks.txt |
| DamageElementEcho | ☐ | 20 | Echo of input damage element information | _DamageElementEcho.csv |

Close

**Step 7** Click **Run** button to launch simulation.

**Step 8** Open the file AER.prn and examine the predicted Average Annual Erosion.

```
Simulation Output
===============================
                          N           Average          SD          Max           Min
Annual Erosion
R:        R-1 -2.0000     1            -2.000         0.000       -2.000        -2.000      1      1
R:        R-2 -2.0000     1            -2.000         0.000       -2.000        -2.000      1      1
R:        R-3 -2.0000     1            -2.000         0.000       -2.000        -2.000      1      1
R:        R-4 -2.0000     1            -2.000         0.000       -2.000        -2.000      1      1
```

Observe that the average rate of shoreline change is identically equal to the assigned Applied Erosion Rate. Note also that the file AER_StormGeneration.csv is empty (no storms generated) and in file AER_StormIteration.csv the reported number of storms is zero. In file AER_MorphologyTimeLine.csv it is seen that at the start of the iteration the berm width is 100 ft and at the end of the iteration the berm width is 0.005, a loss of 99.995 ft in 50 years or an erosion rate of -2.0 ft/year.

Thus, the role of the Applied Erosion Rate has been confirmed.

| Iteration | Time | Reach | UplandWic | DuneHeigh | DuneWidth | BermWidth | EventType | |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | R-1 | 300 | 7.5 | 95 | 100 | INIT | NA |
| 1 | 0 | R-2 | 205 | 16.5 | 45 | 100 | INIT | NA |
| 1 | 0 | R-3 | 205 | 16.5 | 45 | 100 | INIT | NA |
| 1 | 0 | R-4 | 205 | 16.5 | 45 | 100 | INIT | NA |
| 1 | 18256 | R-1 | 300 | 7.5 | 95 | 0.005 | EndIteratio | NA |
| 1 | 18256 | R-2 | 205 | 16.5 | 45 | 0.005 | EndIteratio | NA |
| 1 | 18256 | R-3 | 205 | 16.5 | 45 | 0.005 | EndIteratio | NA |
| 1 | 18256 | R-4 | 205 | 16.5 | 45 | 0.005 | EndIteratio | NA |

# Exercise 11
# Storm-Induced Shoreline Rate of Change

**Step 1** To isolate the storm-induced rate of shoreline change, set the Storm Seasons Probability Active attribute "on" for all seasons. Click *Navigation Tree* item **Storm Seasons**, in the data grid, activate all Storm Seasons by clicking the Probability Active toggle box such that there is a check in each box.

Storm Seasons

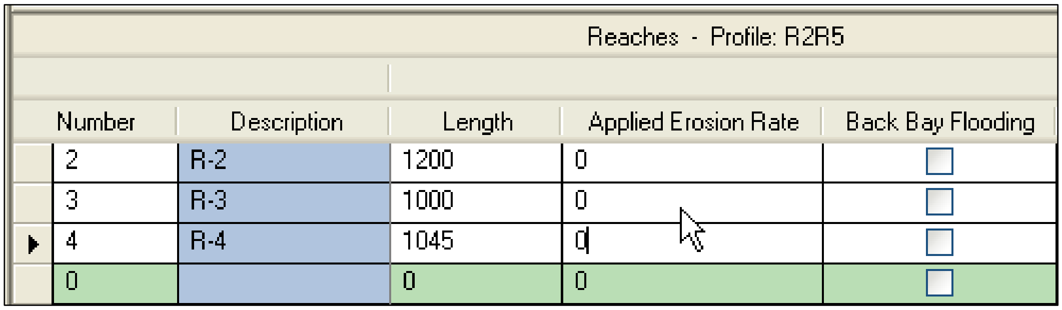
| Number | Description | Minimum Storm Arrival Time | Probability Acitve | Maximum Extra-Tropical Storms |
|---|---|---|---|---|
| 1 | No Storms | 14 | ☑ | 100 |
| 2 | June Storms | 14 | ☑ | 100 |
| 3 | July Storms | 14 | ☑ | 100 |
| 4 | August Storms | 14 | ☑ | 100 |
| 5 | September Storm | 14 | ☑ | 100 |
| 6 | October Storms | 14 | ☑ | 100 |
| 7 | November Storms | 14 | ☑ | 100 |
| 0 | | 0 | ☐ | 100 |

**Step 2** Change the Representative Profile attributes Default Berm Width from 100 to zero (**0**).

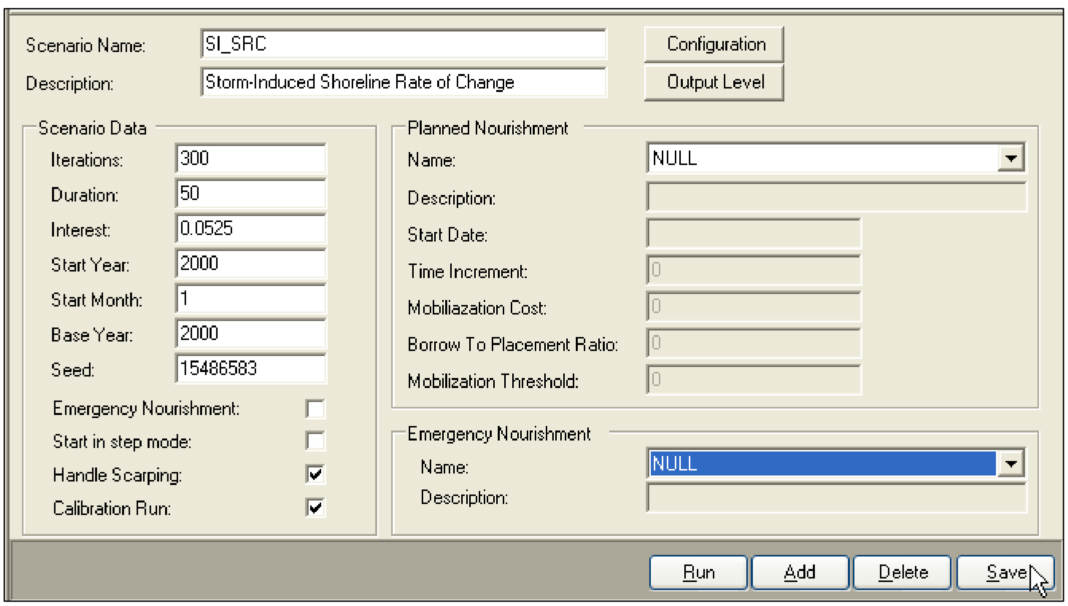
Profiles

| Number | Description | Default Dune Height | Default Dune Width | Default Berm Height | Default Berm Width |
|---|---|---|---|---|---|
| 1 | R1 | 7.5 | 95 | 6.6 | 0 |
| 2 | R2R5 | 16.5 | 45 | 6.6 | 0 |
| 0 | | 0 | 0 | 0 | 0 |

**Step 3** For all Reaches, change the Reach attribute Applied Erosion Rate from -2 to zero (**0**).




**Step 4** Create a new run Scenario for this simulation.




Note that this is a 300-iteration run, in order to obtain behavior associated with variability of the storm sequences and obtain an average storm induced shoreline rate of change. **Save** Scenario and **Run** simulation.

**Step 5** Open the file SI_SRC.prn and examine the predicted Average Annual Erosion.

Simulation Output
================================

| | | | N | Average | SD | Max | Min | | |
|---|---|---|---|---|---|---|---|---|---|
| Annual Erosion | | | | | | | | | |
| R: | R-1 | 0.0000 | 300 | -3.586 | 1.582 | -0.483 | -7.973 | 279 | 53 |
| R: | R-2 | 0.0000 | 300 | -0.752 | 0.469 | 0.000 | -2.903 | 20 | 107 |
| R: | R-3 | 0.0000 | 300 | -0.752 | 0.469 | 0.000 | -2.903 | 20 | 107 |
| R: | R-4 | 0.0000 | 300 | -0.752 | 0.469 | 0.000 | -2.903 | 20 | 107 |

Observe that the average rate of shoreline change is -3.586 feet/year for Reach R-1 and -0.752 ft/year for Reaches R-2, R-3 and R-4. Note also that the files SI_SRC_StormGeneration.csv and SI_SRC_StormIteration.csv both record the occurrence of storms in each of the specified seasons. In file SI_SRC_Iteration.csv note that the moving average number of storms for the 300 iteration simulation is 20.027 storms per 50 year iteration which, when converted to an annual probability of storm occurrence is 0.40, equal to the cumulative probability of storm occurrence (Exercise 2). In file SI_SRC_MorphologyTimeLine.csv it is seen that for Reach R-1, the dune feature is removed completely and upland width is reduced significantly by the end of most iterations. In fact, in a number of iterations the upland width becomes negative (e.g., iterations 53, 63, 71, etc.).

# Exercise 12
# Calibration Simulations

Exercises 10 and 11 have provided two pieces of information that are important for the successful calibration of Beach-*fx*. The first is that in the absence of storms Beach-*fx* will return an average annual rate of shoreline change approximately equal to the Applied Erosion Rate. The second is that for this specific example project the storm climatology and morphology conditions with post-storm recovery set at 90 percent the storm-induced rate of shoreline change is -3.586 and -0.752 ft/year for representative Profiles R1 and R2R5, respectively. The historical shoreline rate of change is as indicated in Table 4 for the four Reaches.

Table 4. Historical shoreline rate of change.

| Reach Name | Shoreline Rate of Change (ft/year) |
|---|---|
| R-1 | -1.7 |
| R-2 | -1.3 |
| R-3 | -0.7 |
| R-4 | 0.0 |

**Step 1** Estimate Applied Erosion Rate (AER) attribute for Reaches R-1, R-2, R-3 and R-4 using the relationship.

AER = Target_HSR - SI_SRC

where

AER = Applied Erosion Rate

Target_HSR = Target Historical Shoreline Rate of Change

SI_SRC = Storm-Induced Shoreline Rate of Change

Knowing the target historical shoreline rate of change and having calculated the storm induced shoreline rate of change, we can estimate the value of AER that should, on average, return the historical rate when combined with the storm-induced changes.

$$\begin{array}{lcrcl}
AER_{R\text{-}1} & = & -\,1.7 & -(-3.586) & = \;\; 1.886 \\
AER_{R\text{-}2} & = & -\,1.3 & -(-0.752) & = -\,0.548 \\
AER_{R\text{-}3} & = & -\,0.7 & -(-0.752) & = \;\; 0.052 \\
AER_{R\text{-}4} & = & 0.0 & -(-0.752) & = \;\; 0.752
\end{array}$$

Specify the estimated AER attribute for each Reach.

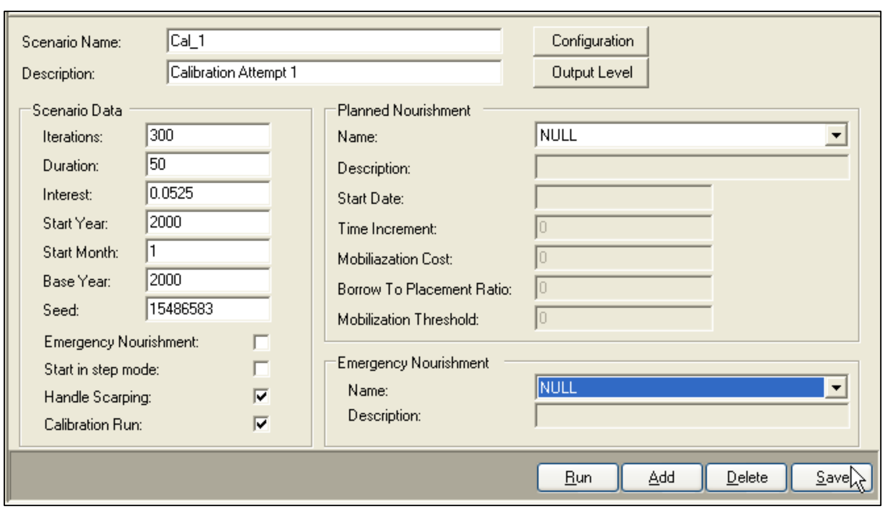
Reaches - Profile: R1

| Number | Description | Length | Applied Erosion Rate | Back Bay Flooding | Emergency Nourishment |
|---|---|---|---|---|---|
| 1 | R-1 | 1600 | 1.886 | ☐ | ☑ |
| 0 | | 0 | 0 | ☐ | ☐ |

Reaches - Profile: R2R5

| Number | Description | Length | Applied Erosion Rate | Back Bay Flooding | Emergency Nourishment |
|---|---|---|---|---|---|
| 2 | R-2 | 1200 | -0.548 | ☐ | ☑ |
| 3 | R-3 | 1000 | 0.052 | ☐ | ☑ |
| 4 | R-4 | 1045 | 0.752 | ☐ | ☑ |
| 0 | | 0 | 0 | ☐ | ☐ |

**Step 2** Define run Scenario Cal_1, a 300-iteration simulation.




**Save** Scenario Cal_1 and **Run** simulation.

**Step 3** Open the file Cal_1.prn and examine the predicted Average Annual Erosion.

Simulation Output

| | | | N | Average | SD | Max | Min | | |
|---|---|---|---|---|---|---|---|---|---|
| Annual Erosion | | | | | | | | | |
| R: | R-1 | 1.8860 | 300 | -1.464 | 1.424 | 1.131 | -5.694 | 176 | 63 |
| R: | R-2 | -0.5480 | 300 | -1.161 | 0.427 | -0.000 | -3.690 | 203 | 107 |
| R: | R-3 | 0.0520 | 300 | -0.684 | 0.472 | 0.000 | -2.903 | 20 | 107 |
| R: | R-4 | 0.7520 | 300 | -0.173 | 0.422 | 0.469 | -2.394 | 203 | 221 |

Observe that the average rate of shoreline change is -1.464 ft/year for Reach R-1, -1.161, -0.684, -0.173 ft/year for Reaches R-2, R-3 and R-4, respectively.

**Step 4** Revise the estimate of Applied Erosion Rate for each Reach using the relationship.

$$AER = AER_{Cal\_1} + Target\_HSR - Cal\_1\_SRC$$

$$AER_{R\text{-}1} = 1.886 + (-1.7) - (-1.464) = 1.650$$
$$AER_{R\text{-}2} = -0.548 + (-1.3) - (-1.161) = -0.687$$
$$AER_{R\text{-}3} = 0.052 + (-0.7) - (-0.684) = 0.036$$
$$AER_{R\text{-}4} = 0.752 + (0.0) - (-0.173) = 0.925$$

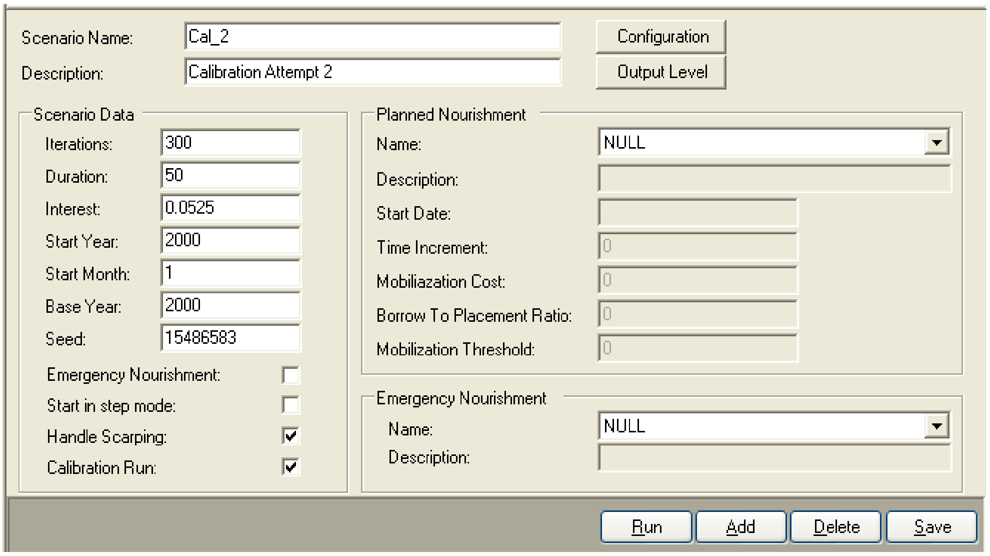
Reaches - Profile: R1

| Number | Description | Length | Applied Erosion Rate | Back Bay Flooding | Emergency Nourishment |
|---|---|---|---|---|---|
| 1 | R-1 | 1600 | 1.650 | ☐ | ☑ |
| 0 | | 0 | 0 | ☐ | ☐ |

Reaches - Profile: R2R5

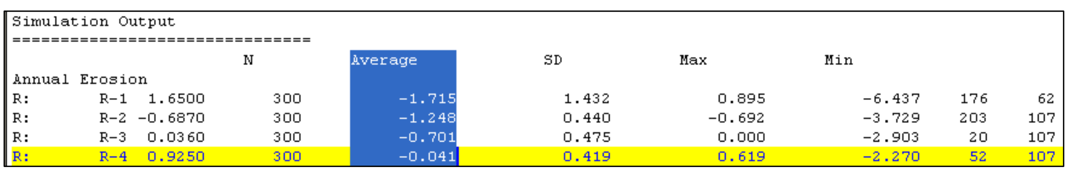
| Number | Description | Length | Applied Erosion Rate | Back Bay Flooding | Emergency Nourishment |
|---|---|---|---|---|---|
| 2 | R-2 | 1200 | -0.687 | ☐ | ☑ |
| 3 | R-3 | 1000 | 0.036 | ☐ | ☑ |
| 4 | R-4 | 1045 | 0.925 | ☐ | ☑ |
| 0 | | 0 | 0 | ☐ | ☐ |

Define run Scenario Cal_2




**Save** Scenario Cal_2 and **Run** simulation.

**Step 5** Open the file Cal_2.prn and examine the predicted Average Annual Erosion.


Simulation Output
===============================

| | | | N | Average | SD | Max | Min | | |
|---|---|---|---|---|---|---|---|---|---|
| Annual Erosion | | | | | | | | | |
| R: | R-1 | 1.6500 | 300 | -1.715 | 1.432 | 0.895 | -6.437 | 176 | 62 |
| R: | R-2 | -0.6870 | 300 | -1.248 | 0.440 | -0.692 | -3.729 | 203 | 107 |
| R: | R-3 | 0.0360 | 300 | -0.701 | 0.475 | 0.000 | -2.903 | 20 | 107 |
| R: | R-4 | 0.9250 | 300 | -0.041 | 0.419 | 0.619 | -2.270 | 52 | 107 |

The predicted average rate of shoreline change from this simulation is close to the target historical erosion rate, in fact, Reach R-3 is calibrated. Continuing the procedure described in Step 4 for two or three more simulations will result in a precise calibration with the Applied Erosion Rates listed in Table 5 which is considered sufficient for calibration.

Table 5. Example project calibration settings.

| Reach Name | Shoreline Rate of Change (ft/year) | Applied Erosion Rate (ft/year) |
|---|---|---|
| R-1 | -1.7 | 1.665 |
| R-2 | -1.3 | -0.764 |
| R-3 | -0.7 | 0.036 |
| R-4 | 0.0 | 0.975 |

## Calibration Discussion

Exercises 10 through 12 outline the general approach for calibration of Beach-*fx*. Recall that calibration of Beach-*fx* is a balance between the interrelated input specifications of storm climatology, post-storm berm width recovery and the applied erosion rate to arrive at an uncertain quantity referred to as the long-term historical shoreline rate of change. In the calibration procedure outlined, the applied erosion rate is treated as the only free variable, whereas in truth, post-storm berm width recovery is of unknown magnitude and for that matter so also is the storm climatology and historical shoreline rate of change but perhaps to a lesser degree. Because of this, the coastal specialist in the project team must give careful consideration to the assignment of the post-storm berm width recovery specification and recognize that alternative specifications will result in a different set of calibration values. From a philosophical perspective, the applied erosion rate would represent the long-term shoreline rate of change at the project site in the absence of the significant storms included in the plausible storm suite. For the example project studied here, changing the post-storm berm width recovery specification from 90 percent to 80 percent results in a 0.614 to 1.275 ft/year change in the resulting applied erosion rate specification as illustrated in Table 6.

Table 6. Applied erosion rate sensitivity to post-storm berm width recovery (ft/year).

| Reach Name | Berm Width Recovery | |
|---|---|---|
| | 90% | 80% |
| R-1 | 1.665 | 2.94 |
| R-2 | -0.764 | -0.144 |
| R-3 | 0.036 | 0.650 |
| R-4 | 0.975 | 1.670 |

# Training Exercises

# Part III

# Without-Project Simulations

# Exercise 13
# Define Emergency Nourishment Alternative

The present implementation of Emergency Nourishment Alternatives within Beach-*fx* is specification of a nourishment fill density (cubic yards/feet) that acts to increase dune width (at the current dune elevation) at the expense of berm width. However, if the current beach morphology is in a scarping condition the fill material is first used to restore the berm for the deficit volume represented by the scarping condition. It is possible that the scarping-induced volume deficit is greater than the specified Emergency Nourishment fill density in which case the scarping condition is reduced but not entirely restored and the dune width remains unchanged. The model development team continues to research and develop other methods for implementing Emergency Nourishment including; Emergency Nourishment to increase dune elevation and dune width based on a fill density specification and Emergency Nourishment to a user-specified morphology template, independent of a fill density specification. As a result, entry of Emergency Nourishment Alternative specifications is likely to change in future releases of Beach-*fx*.

**Step 1** Click on the *Navigation Tree* item **Configuration Settings**. There are four Emergency Nourishment control specifications contained in the Configuration Settings.

- EmergencyNourishmentFillDensitySpecification
- EmergencyNourishmentTriggerAdjustment
- EmergencyNourishmentMobilizationTimeThreshold
- EmergencyNourishmentScheduledNourishment BlackoutWindowMultiplier

The EmergencyNourishmentFillDensitySpecification is the specified Emergency Nourishment fill density (cubic yards/feet). The EmergencyNourishmentTriggerAdjustment is the additional dune width loss (feet) that will trigger a subsequent Emergency Nourishment after construction of the present Emergency Nourishment. The EmergencyNourishmentMobilizationTime Threshold is a time window specification (days) beyond which any subsequent Emergency

Nourishment will incur another project level mobilization cost. The EmergencyNourishment ScheduledNourishmentBlackoutWindowMultiplier is a multiplier (factor) that is applied to the Emergency Nourishment mobilization time and checked against the start date of any Planned Nourishment activity. If a Planned Nourishment activity is scheduled to begin on or before the calculated date then the Emergency Nourishment activity is canceled. This is a necessary scheduling issue that will prohibit an overlapping of Emergency Nourishment and Planned Nourishment activities. From a coastal management perspective, it is not likely that Emergency Nourishment would be constructed just prior to a scheduled comprehensive nourishment project.

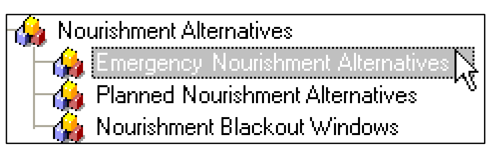
Assign the highlighted Emergency Nourishment specifications in the Configuration Settings table. Note that these specifications are global to the project, and do not vary by Reach.

- EmergencyNourishmentFillDensitySpecification (6)
- EmergencyNourishmentTriggerAdjustment (1)
- EmergencyNourishmentMobilizationTimeThreshold (20)
- EmergencyNourishmentScheduledNourishmentBlackoutWindowMultiplier (2)

Configuration Settings

| Tag | Description | Value |
|---|---|---|
| UsePlanform | if >=0 apply stored planform rates, if < 0 planform = 0 all reaches | 1 |
| ToleranceNoProfileChange | tolerance for assume no profile change (de minimis) | 0.1 |
| TimeStepReportInterval | Report every nth time step (0 = no report, 1 = every time step, 2= ever | 10 |
| TimeStep | Time step for planform / historical adjustments, days | 7 |
| StormRecoveryPeriod | Time in days after storm for recovery to be applied | 21 |
| StormGenerationOnly | Storm Generation Run Only if > 0 | 0 |
| NoWriteToDataBase | Turn off writing to output data base if > 0 (used for testing) | 0 |
| MaximumRecoverableScarping | Maximum amount of scarping that can be recovered (feet) | 5 |
| LotArmoringInEffect | Lot Armoring In Effect if > 0 | -1 |
| LookupDuneWidthExceptionReportToleranc | Value in feet for flagging exception report on lookups for dune width | 20 |
| LookupDuneHeightExceptionReportToleran | Value in feet for flagging exception report on lookups for dune height | 20 |
| LookupBermWidthExceptionReportToleranc | Value in feet for flagging exception report on lookups for berm width | 20 |
| EmergencyNourishmentTriggerAdjustment | Emergency Nourishment Trigger Adjustment > 0 set trigger greater th | 1 |
| EmergencyNourishmentScheduledNourishm | Multiplier on reach emegency nourishment mobilization to set blackou | 2 |
| EmergencyNourishmentMobilizationTimeThr | Time Window after end of last Emergency Nourishment at which mobi | 20 |
| EmergencyNourishmentFillDensitySpecificat | Cubic Yards/Foot Fill Density For Emergency Nourishment | 6 |
| DepthOfClosureMultiplier | Multiplier on entered depth of closure for use in morphology change c | 1 |
| DataCheckMinimumLotSize | Minimum Lot Size, in acres, for data check flag (lots below this size wi | 0.1 |
| DataCheckMaximumLotSize | Maximum Lot Size, in acres, for data check flag (lots above this size | 0.1 |
| DamageElementDistanceTolerance | Tolerance (feet) for damage element distance checks | 10 |
| CondemnationRatio | global condemnation ratio | 0.5 |
| BarrowFloodingReductionThreshold | If adjusted Barrow value < Threshold then set flooding value to 0 | 0 |
| BarrowFloodingReductionMultiplier | Amount to multiply the returned water level in Barrow flooding approa | 1 |
| BarrowFloodingReductionAmount | Amount to reduce the returned water level in Barrow flooding approac | 0 |
| BarrowFloodingFlag | Use Barrow Flooding Approach if > 0 | -1 |

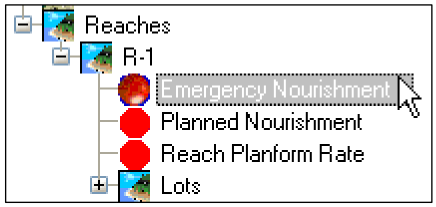
**Step 2** Expand *Navigation Tree* item Nourishment Alternatives, click on **Emergency Nourishment Alternatives**. In the data grid, provide a Name, Description, and specify the project level mobilization cost for Emergency Nourishment operations.




Emergency Nourishment Alternatives

| Name | Description | Mobilization Cost |
|---|---|---|
| NULL | | 0 |
| WoP_EN | Example Project Emergency Nourishment | 15000 |
| | | 0 |

**Step 3** Expand Reach R-1, click on **Emergency Nourishment**.




In the data grid select alternative **WoP_EN** from the pull-down, enter a unit placement cost ($/cubic yards) of **30**, a borrow to placement ratio of **1**, a production rate (cubic yards/day) of **1,500**, a Reach level mobilization cost of **0**, a mobilization time (days) of **90**, priority order **1** (this determines which Reach receives nourishment first, if more than one Reach is triggered for Emergency Nourishment at the same time), dune height trigger of **0** (inactive attribute), dune width trigger of **90** (feet), berm width trigger **0** (inactive attribute), leave the Emergency Template attributes blank (these attributes are inactive).

| Emergency Nourishment Alternatives | | | | | |
|---|---|---|---|---|---|
| Emergency Nourishment Alternative | Unit Placement Cost | Borrow To Placement Ratio | Production Rate | Mobilization Cost | Mobilization Time |
| WoP_EN | 30 | 1 | 1500 | 0 | 90 |
| 0 | 0 | 0 | 0 | | |

| Emergency Nourishment Alternatives | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | Emergency Nourishment Trigger | | | Emergency Template | | |
| Emergency Nourishment Alternative | Priority Order | Dune Height | Dune Width | Berm Width | Dune Height | Dune Width | Berm Width |
| WoP_EN | 1 | 0 | 90 | 0 | | | |
| 0 | 0 | 0 | 0 | 0 | | | |

Similar specifications need to be provided for Reaches R-2, R-3, and R-4. All of the attributes are the same except the Priority Order attribute, which should be **2**, **3**, and **4** for Reaches R-2, R-3, and R-4, respectively. Also the Emergency Nourishment Trigger for Dune Width should be **44** for Reaches R-2, R-3, and R-4. After entering these specifications at the Reach level, the full specification can be viewed by

clicking on the *Navigation Tree* item Emergency Nourishment and the data table should contain values as illustrated below.

| Emergency Nourishment Alternatives | | | | | | |
|---|---|---|---|---|---|---|
| Reach | Emergency Nourishment Alternative | Unit Placement Cost | Borrow To Placement Ratio | Production Rate | Mobilization Cost | Mobilization Time |
| R-1 | WoP_EN | 30 | 1 | 1500 | 0 | 90 |
| R-2 | WoP_EN | 30 | 1 | 1500 | 0 | 90 |
| R-3 | WoP_EN | 30 | 1 | 1500 | 0 | 90 |
| R-4 | WoP_EN | 30 | 1 | 1500 | 0 | 90 |

| Emergency Nourishment Alternatives | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Emergency Nourishment Trigger | | | Emergency Template | | |
| Reach | Emergency Nourishment Alternative | Priority Order | Dune Height | Dune Width | Berm Width | Dune Height | Dune Width | Berm Width |
| R-1 | WoP_EN | 1 | 0 | 90 | 0 | | | |
| R-2 | WoP_EN | 2 | 0 | 44 | 0 | | | |
| R-3 | WoP_EN | 3 | 0 | 44 | 0 | | | |
| R-4 | WoP_EN | 4 | 0 | 44 | 0 | | | |

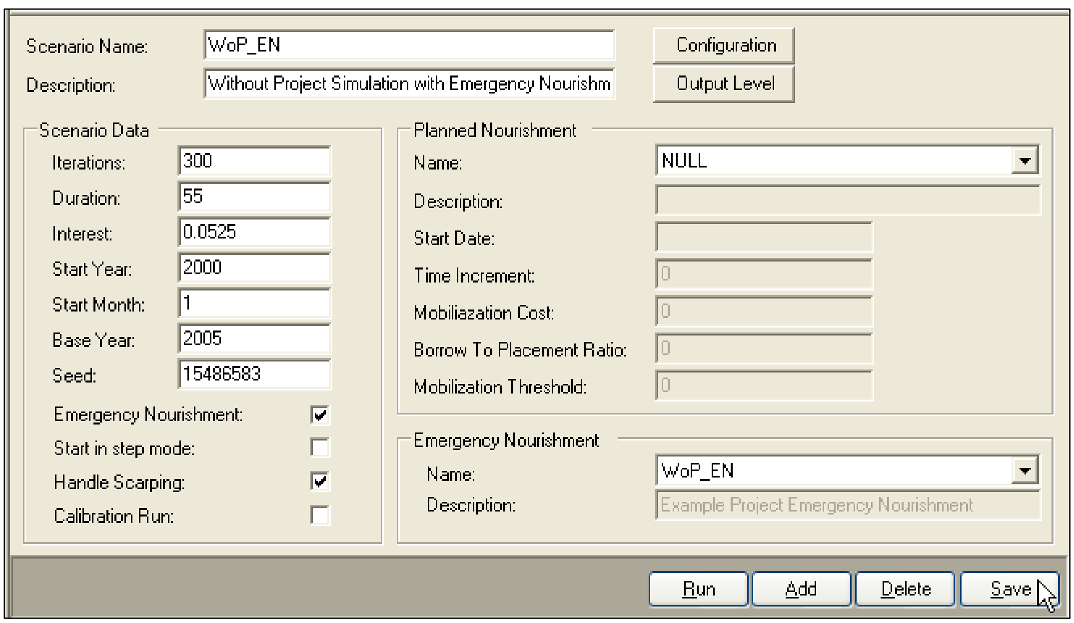
**Step 4** Confirm that the Emergency Nourishment toggle (a Reach level attribute) is checked by clicking on the navigation tree items R-1, R-2, R-3, and R-4.

| Reaches - Profile: R1 | | | | | |
|---|---|---|---|---|---|
| Number | Description | Length | Applied Erosion Rate | Back Bay Flooding | Emergency Nourishment |
| 1 | R-1 | 1600 | 1.665 | ☐ | ☑ |
| 0 | | 0 | 0 | ☐ | ☐ |

| Reaches - Profile: R2R5 | | | | | |
|---|---|---|---|---|---|
| Number | Description | Length | Applied Erosion Rate | Back Bay Flooding | Emergency Nourishment |
| 2 | R-2 | 1200 | -0.765 | ☐ | ☑ |
| 3 | R-3 | 1000 | 0.036 | ☐ | ☑ |
| 4 | R-4 | 1045 | 0.975 | ☐ | ☑ |
| 0 | | 0 | 0 | ☐ | ☐ |

# Exercise 14
# Define Without-Project Simulation(s)

**Step 1** Define run Scenario for the Without-Project Simulation. Scenario Name: **WoP_EN**, Description: **Without-Project Simulation with Emergency Nourishment**. Scenario Data: Iterations = **300**, Duration = **55**, Interest = **0.0525**, Start Year = **2,000**, Start Month = **1**, Base Year = **2005**, Seed = **15486583**, Emergency Nourishment = **checked**, Start in Step Mode = **blank**, Handle Scarping = **checked**, Calibration Run = **blank**. Planned Nourishment: select **NULL** from pull-down. Emergency Nourishment: select **WoP_EN**.

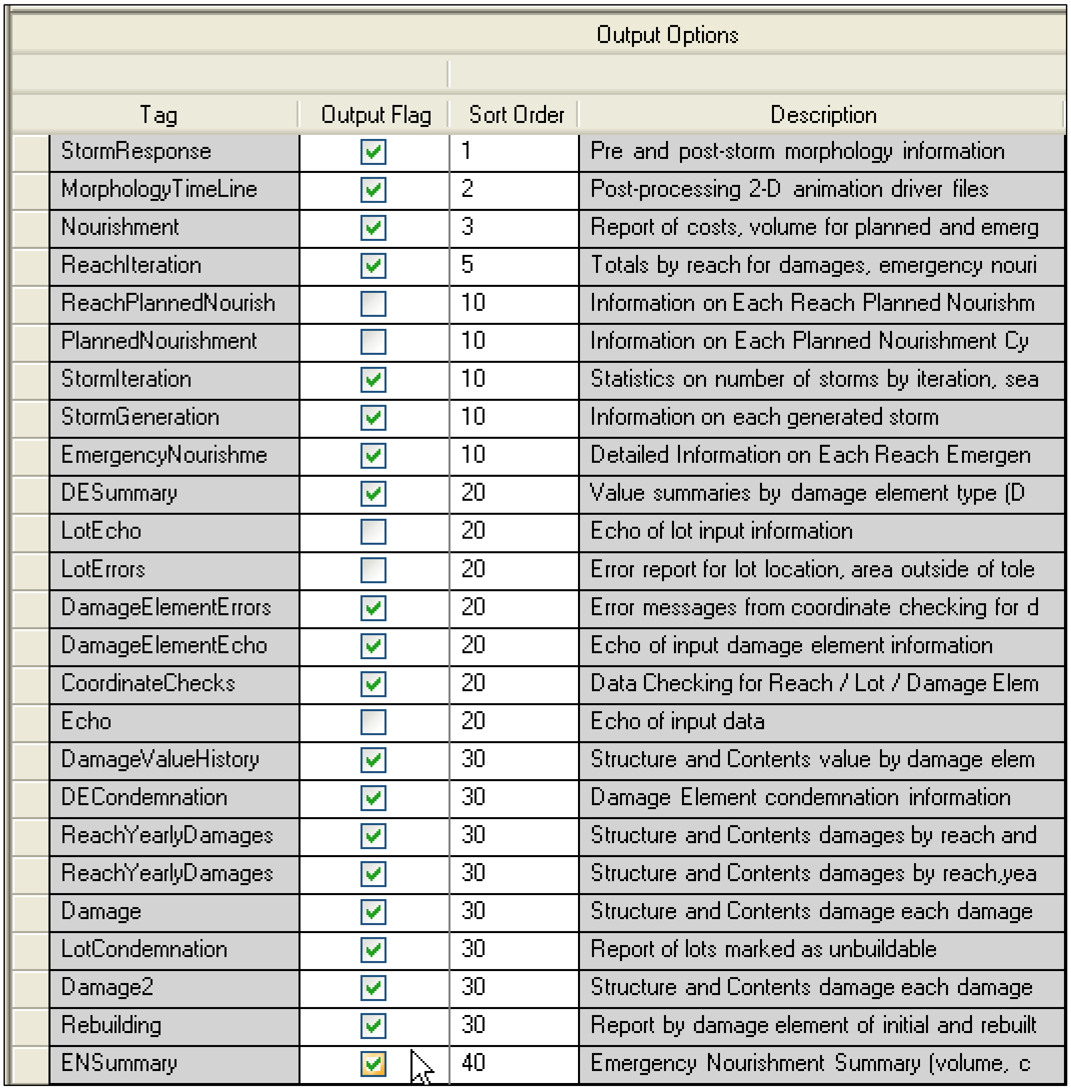



Click the **Output Level** button to check activated output files. Turn on output for:

- StormResponse
- MorphologyTimeLine
- Nourishment
- ReachIteration
- StormIteration
- StormGeneration

- EmergencyNourishment
- DESummary
- DamageElementErrors
- DamageElementEcho
- CoordinateChecks
- DamageValueHistory
- DECondemnation
- ReachYearlyDamages
- Damage
- LotCondemnation
- Damage2
- Rebuilding
- ENSummary

This is a rather exhaustive list of requested output files and it is not likely that you will require all of these output files for every simulation. However, it is important to understand the types of output available in each of the output files and to check for geographical positioning errors related to Damage Elements, Lots, and Reaches at least once in all projects.



| Output Options | | | |
|---|---|---|---|
| Tag | Output Flag | Sort Order | Description |
| StormResponse | ☑ | 1 | Pre and post-storm morphology information |
| MorphologyTimeLine | ☑ | 2 | Post-processing 2-D animation driver files |
| Nourishment | ☑ | 3 | Report of costs, volume for planned and emerg |
| ReachIteration | ☑ | 5 | Totals by reach for damages, emergency nouri |
| ReachPlannedNourish | ☐ | 10 | Information on Each Reach Planned Nourishm |
| PlannedNourishment | ☐ | 10 | Information on Each Planned Nourishment Cy |
| StormIteration | ☑ | 10 | Statistics on number of storms by iteration, sea |
| StormGeneration | ☑ | 10 | Information on each generated storm |
| EmergencyNourishme | ☑ | 10 | Detailed Information on Each Reach Emergen |
| DESummary | ☑ | 20 | Value summaries by damage element type (D |
| LotEcho | ☐ | 20 | Echo of lot input information |
| LotErrors | ☐ | 20 | Error report for lot location, area outside of tole |
| DamageElementErrors | ☑ | 20 | Error messages from coordinate checking for d |
| DamageElementEcho | ☑ | 20 | Echo of input damage element information |
| CoordinateChecks | ☑ | 20 | Data Checking for Reach / Lot / Damage Elem |
| Echo | ☐ | 20 | Echo of input data |
| DamageValueHistory | ☑ | 30 | Structure and Contents value by damage elem |
| DECondemnation | ☑ | 30 | Damage Element condemnation information |
| ReachYearlyDamages | ☑ | 30 | Structure and Contents damages by reach and |
| ReachYearlyDamages | ☑ | 30 | Structure and Contents damages by reach,yea |
| Damage | ☑ | 30 | Structure and Contents damage each damage |
| LotCondemnation | ☑ | 30 | Report of lots marked as unbuildable |
| Damage2 | ☑ | 30 | Structure and Contents damage each damage |
| Rebuilding | ☑ | 30 | Report by damage element of initial and rebuilt |
| ENSummary | ☑ | 40 | Emergency Nourishment Summary (volume, c |

Click **Save** to save the Scenario specifications and **Run** to launch the simulation.

# Training Exercises

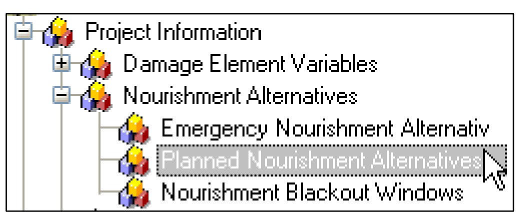
# Part IV

# With-Project Simulations

# Exercise 15
# Define Planned Nourishment Alternative

The present implementation of Planned Nourishment Alternatives within Beach-*fx* termed a periodic-tested scheme involves specification of a nourishment template and threshold trigger specifications (expressed as a percent of template values) along with a target renourishment interval, start date, mobilization threshold and mobilization costs. On the start date, the required nourishment volume is estimated for all Reaches in which at least one of the threshold trigger specifications is satisfied. If the required nourishment volume exceeds the mobilization threshold volume then a Planned Nourishment activity is scheduled. When nourishment occurs all Reaches are restored to the specified nourishment template regardless of the nourishment threshold triggers. Renourishments are processed in a similar manner at the specified renourishment interval. If the mobilization threshold volume is not exceeded nourishment does not take place, the next Planned Nourishment check occurs at the end of the renouishment interval. As with the implementation of Emergency Nourishment the model development team continues to research and develop other methods for implementing Planned Nourishment including; periodic-forced, mobilization-tested, budget-tested and adaptive nourishment schemes. As a result, entry of Planned Nourishment Alternative specifications is likely to change in future releases of Beach-*fx*.

**Step 1** Click on the *Navigation Tree* item **Planned Nourishment Alternatives**.




In the data grid provide a Name (**Alt_1**), Description (**Alternative 1 Nourishment**), Start Date (**08/01/2005**), Time Increment (**7**), Mobilization Cost (**400,000**), Default Borrow To Placement Ratio (**1**), Mobilization Threshold (**500,000**) and Planned Nourishment implementation Type (**Periodic-Tested**).

Planned Nourishment Alternatives

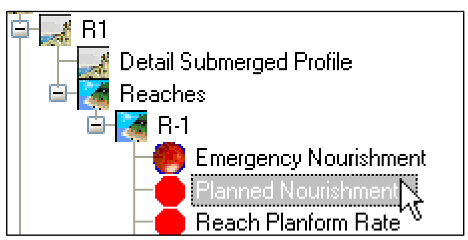
| Name | Description | Start Date | Time Increment | Mobilization Cost |
|---|---|---|---|---|
| Alt_1 | Alternative 1 Nourishment | 08/01/2005 | 7 | 400000 |
| NULL | | | 0 | 0 |
| | | | 0 | 0 |

Planned Nourishment Alternatives

| Name | Description | Default Borrow To Placement Ratio | Mobilization Threshold | Type |
|---|---|---|---|---|
| Alt_1 | Alternative 1 Nourishment | 1 | 500000 | Periodic-Tested |
| NULL | | 0 | 0 | Periodic-Tested |
| | | 0 | 0 | Periodic-Tested |

This step has defined the project level Planned Nourishment plan. The next step involves providing the Reach level Planned Nourishment specifications.

**Step 2** Click on the *Navigation Tree* item **Planned Nourishmen**t under Reach R-1. This opens the Reach level Planned Nourishment data grid for Reach R-1.

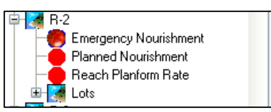



In the data grid, select Planned Nourishment Alternative **Alt_1** from the pull-down menu, specify a unit placement cost of **5** ($/cubic yards), Borrow To Placement Ratio of **1**, Production Rate of **30,000** (cubic yards/day), Processing Order of **1**, Planned Nourishment Trigger values of **0.5** for Berm Width, **0.6** for Dune Width, and **0.8** for Dune Height, Template values of 9**.5** for Dune Height, **99** for Dune Width and **100** for Berm width. This completes the specification of the Planned Nourishment for Reach R-1. Similar specifications must be provided for Reaches R-2, R-3, and R-4.

| Planned Nourishment Alternatives | | | | |
|---|---|---|---|---|
| Planned Nourishment Alternative | Unit Placement Cost | Borrow To Placement Ratio | Production Rate | Processing Order |
| Alt_1 | 5 | 1 | 30000 | 1 |
| 0 | 0 | 0 | 0 | 0 |

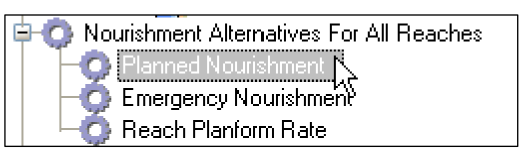
| Planned Nourishment Alternatives | | | | | | |
|---|---|---|---|---|---|---|
| | Planned Nourishment Trigger | | | Template | | |
| Planned Nourishment Alternative | Berm Width | Dune Width | Dune Height | Dune Height | Dune Width | Berm Width |
| Alt_1 | 0.5 | 0.6 | 0.8 | 9.5 | 99 | 100 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

In Reaches R-2, R-3, and R-4 the Unit Placement costs, Borrow To Placement Ratio and Production Rate specifications are the same as for Reach R-1. The Processing Order is **2**, **3**, and **4**, for Reaches R-2, R-3, and R-4, respectively. The Planned Nourishment Trigger values are **0.5** for Berm Width, **0.6** for Dune Width and **0.9** for Dune Height. The Template values are **18.5** for Dune Height, **55** for Dune Width and **100** for Berm Width.




| Planned Nourishment Alternatives | | | | | | |
|---|---|---|---|---|---|---|
| | Planned Nourishment Trigger | | | Template | | |
| Planned Nourishment Alternative | Berm Width | Dune Width | Dune Height | Dune Height | Dune Width | Berm Width |
| Alt_1 | 0.5 | 0.6 | 0.9 | 18.5 | 55 | 100 |

After completing the Planned Nourishment specifications for each of the four Reaches a summary of the full specification can be viewed by clicking on the *Navigation Tree* item **Planned Nourishment** under Nourishment Alternatives For All Reaches. The values in this table should be as illustrated below.



| Planned Nourishment Alternatives | | | | | |
|---|---|---|---|---|---|
| Reach | Planned Nourishment Alternative | Unit Placement Cost | Borrow To Placement Ratio | Production Rate | Processing Order |
| R-1 | Alt_1 | 5 | 1 | 30000 | 1 |
| R-2 | Alt_1 | 5 | 1 | 30000 | 2 |
| R-3 | Alt_1 | 5 | 1 | 30000 | 3 |
| R-4 | Alt_1 | 5 | 1 | 30000 | 4 |

| Planned Nourishment Alternatives | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | Planned Nourishment Trigger | | | Template | | |
| Reach | Planned Nourishment Alternative | Berm Width | Dune Width | Dune Height | Template Dune Height | Template Dune Width | Template Berm Width |
| R-1 | Alt_1 | 0.5 | 0.6 | 0.8 | 9.5 | 99 | 100 |
| R-2 | Alt_1 | 0.5 | 0.6 | 0.9 | 18.5 | 55 | 100 |
| R-3 | Alt_1 | 0.5 | 0.6 | 0.9 | 18.5 | 55 | 100 |
| R-4 | Alt_1 | 0.5 | 0.6 | 0.9 | 18.5 | 55 | 100 |

**Step 3** The final step in completing the Planned Nourishment specification is to provide, at the Reach level, estimates of the project-induced shoreline rate of change. This planform rate of change specification allows Beach-*fx* to capture the dispersion of the nourishment project (end losses). The source of these data specifications are the GENESIS simulations of the project's evolution. Specifically, it is the difference between the with- and without-project simulations of the project's evolution expressed as an annual shoreline rate of change by nourishment cycle. The estimated project-induced shoreline rate of change for the example project is listed in Table 7.

Table 7. Project-induced shoreline rate of change (feet/year).

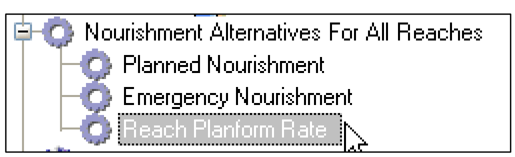
| Nourishment Cycle | Reach Name | | | |
|---|---|---|---|---|
| | R-1 | R-2 | R-3 | R-4 |
| 1 | -7.0 | -6.2 | -6.6 | -7.2 |
| 2 | -4.5 | -3.9 | -4.2 | -4.6 |
| 3 | -3.5 | -3.1 | -3.3 | -3.6 |
| 4 | -2.8 | -2.4 | -2.6 | -2.9 |
| 5 | -2.7 | -2.3 | -2.5 | -2.8 |
| 6 | -2.3 | -2.0 | -2.2 | -2.4 |
| 7 | -2.2 | -1.9 | -2.1 | -2.3 |

Click on the *Navigation Tree* item **Reach Planform Rate** under Reach R-1, in the data grid select Planned Nourishment Alternative **Alt_1** from the pull-down menu and specify the nourishment cycle and corresponding project-induced shoreline rate of change as illustrated in the following image. Repeat this process for Reaches R-2, R-3, and R-4.

Reach Planform Rate

| Planned Nourishment Alternative | Nourishment Cycle | Berm Width Change Rate |
|---|---|---|
| Alt_1 | 1 | -7 |
| Alt_1 | 2 | -4.5 |
| Alt_1 | 3 | -3.5 |
| Alt_1 | 4 | -2.8 |
| Alt_1 | 5 | -2.7 |
| Alt_1 | 6 | -2.3 |
| Alt_1 | 7 | -2.2 |
| | | |

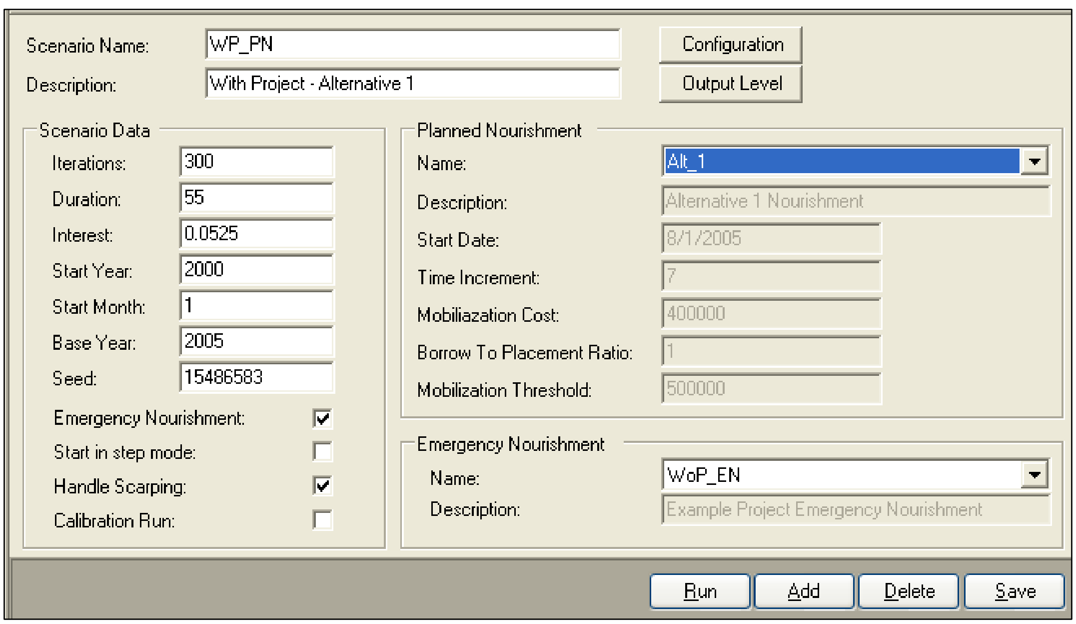
After completing the specification of the Reach Planform Rate for each of the four Reaches a summary of the full specification can be viewed by clicking on the *Navigation Tree* item **Reach Planform Rate** under Nourishment Alternatives For All Reaches. The values in this table should be as illustrated below.



| Reach Planform Rates | | | |
|---|---|---|---|
| Reach | Planned Nourishment Alternative | Nourishment Cycle | Berm Width Change Rate |
| R-1 | Alt_1 | 1 | -7 |
| R-1 | Alt_1 | 2 | -4.5 |
| R-1 | Alt_1 | 3 | -3.5 |
| R-1 | Alt_1 | 4 | -2.8 |
| R-1 | Alt_1 | 5 | -2.7 |
| R-1 | Alt_1 | 6 | -2.3 |
| R-1 | Alt_1 | 7 | -2.2 |
| R-2 | Alt_1 | 1 | -6.2 |
| R-2 | Alt_1 | 2 | -3.9 |
| R-2 | Alt_1 | 3 | -3.1 |
| R-2 | Alt_1 | 4 | -2.4 |
| R-2 | Alt_1 | 5 | -2.3 |
| R-2 | Alt_1 | 6 | -2 |
| R-2 | Alt_1 | 7 | -1.9 |

# Exercise 16
# Define With-Project Simulation(s)

**Step 1** Define run Scenario for the With-Project Simulation. Scenario Name: **WP_PN**, Description: **With-Project – Alternative 1**. Scenario Data: Iterations = **300**, Duration = **55**, Interest = **0.0525**, Start Year = **2000**, Start Month = **1**, Base Year = **2005**, Seed = **15486583**, Emergency Nourishment = **checked**, Start in Step Mode = **blank**, Handle Scarping = **checked**, Calibration Run = **blank**. Planned Nourishment: select **Alt_1** from pull-down. Emergency Nourishment: select **WoP_EN**.




Click **Save** to save the Scenario specifications. Additional output files are available for Planned Nourishment. Add files “PlannedNourishment” and “ReachPlannedNourishment” to the enabled output files. Toggle the output files on or off as indicated below:

- StormResponse (off)
- MorphologyTimeLine (on)
- Nourishment (on)
- ReachIteration (on)
- StormIteration (off)

- StormGeneration (off)
- EmergencyNourishment (on)
- DESummary (off)
- DamageElementErrors (off)
- DamageElementEcho (off)
- CoordinateChecks (off)
- DamageValueHistory (on)
- DECondemnation (off)
- ReachYearlyDamages (on)
- Damage (on)
- LotCondemnation (off)
- Damage2 (off)
- Rebuilding (on)
- ENSummary (on)

Output Options

| Tag | Output Flag | Sort Order | Description |
|---|---|---|---|
| StormResponse | ☐ | 1 | Pre and post-storm morphology information |
| MorphologyTimeLine | ☑ | 2 | Post-processing 2-D animation driver files |
| Nourishment | ☑ | 3 | Report of costs, volume for planned and emergency nourishme |
| ReachIteration | ☑ | 5 | Totals by reach for damages, emergency nourishment, profile l |
| EmergencyNourishme | ☑ | 10 | Detailed Information on Each Reach Emergency Nourishment |
| ReachPlannedNourish | ☐ | 10 | Information on Each Reach Planned Nourishment |
| PlannedNourishment | ☐ | 10 | Information on Each Planned Nourishment Cycle |
| StormIteration | ☐ | 10 | Statistics on number of storms by iteration, season, storm type |
| StormGeneration | ☐ | 10 | Information on each generated storm |
| NourishmentCalendar | ☐ | 10 | Start and Finish of Planned and Emergency plus blackout wind |
| DESummary | ☐ | 20 | Value summaries by damage element type (DESummary1) and |
| LotEcho | ☐ | 20 | Echo of lot input information |
| LotErrors | ☐ | 20 | Error report for lot location, area outside of tolerance limits |
| DamageElementErrors | ☐ | 20 | Error messages from coordinate checking for damage elements |
| DamageElementEcho | ☐ | 20 | Echo of input damage element information |
| CoordinateChecks | ☐ | 20 | Data Checking for Reach / Lot / Damage Element Coordinates |
| Echo | ☐ | 20 | Echo of input data |
| DamageValueHistory | ☑ | 30 | Structure and Contents value by damage element over time |
| DECondemnation | ☐ | 30 | Damage Element condemnation information |
| ReachYearlyDamages | ☑ | 30 | Structure and Contents damages by reach and year |
| ReachYearlyDamages | ☑ | 30 | Structure and Contents damages by reach,year, damage eleme |
| Damage | ☑ | 30 | Structure and Contents damage each damage occurrence |
| LotCondemnation | ☐ | 30 | Report of lots marked as unbuildable |
| Damage2 | ☐ | 30 | Structure and Contents damage each damage occurrence alter |
| Rebuilding | ☑ | 30 | Report by damage element of initial and rebuilt structure and co |
| ENSummary | ☑ | 40 | Emergency Nourishment Summary (volume, cost) by reach |

Click **Run** to launch the simulation.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| | | |
|---|---|---|
| **1. REPORT DATE** *(DD-MM-YYYY)* August 2009 | **2. REPORT TYPE** Final Report | **3. DATES COVERED** *(From - To)* |
| **4. TITLE AND SUBTITLE** Beach-*fx* User's Manual: Version 1.0 | | **5a. CONTRACT NUMBER** |
| | | **5b. GRANT NUMBER** |
| | | **5c. PROGRAM ELEMENT NUMBER** |
| **6. AUTHOR(S)** Cory M. Rogers, Kristine A. Jacobson, and Mark B. Gravens | | **5d. PROJECT NUMBER** |
| | | **5e. TASK NUMBER** |
| | | **5f. WORK UNIT NUMBER** |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** CDM, 1050 N. Reed Station Road, Carbondale, IL 62902-7377<br>U.S. Army Engineer Research and Development Center, Coastal and Hydraulics Laboratory 3909 Halls Ferry Road, Vicksburg, MS 39180-6199 | | **8. PERFORMING ORGANIZATION REPORT NUMBER** ERDC/CHL SR-09-6 |
| **9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)** U.S. Army Corps of Engineers<br>Washington, DC 20314-1000 | | **10. SPONSOR/MONITOR'S ACRONYM(S)** |
| | | **11. SPONSOR/MONITOR'S REPORT NUMBER(S)** |

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

The need to strengthen the linkages between engineering analyses (project performance and evolution) and planning functions (alternative analysis and economic justification) with respect to coastal storm damage reduction projects within the Corps has led to the development of the life-cycle simulation model Beach-*fx*. Beach-*fx* provides a comprehensive analytical framework for evaluating the physical performance and economic benefits and costs of shore protection projects, particularly beach nourishment along sandy shores. The model has been implemented as an event-based Monte Carlo life-cycle simulation tool that is run on desktop computers. This report describes the components, purpose, and operational function of the Beach-*fx* graphical user interface, including navigation within the interface and the organization and specification of all model input and output data.

**15. SUBJECT TERMS**

| | | |
|---|---|---|
| Shore protection projects | Life-cycle simulation | Economic benefits and costs |
| Engineering – economic analysis | Physical performance | Beach nourishment |

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** |
|---|---|---|---|---|---|
| **a. REPORT** UNCLASSIFIED | **b. ABSTRACT** UNCLASSIFIED | **c. THIS PAGE** UNCLASSIFIED | | 261 | **19b. TELEPHONE NUMBER** *(include area code)* |